AFRL-HE-WP-TR-2002-0039

UNITED STATES AIR FORCE RESEARCH LABORATORY

Aircraft Battle Damage Assessment and Repair (ABDAR) Final Program Report

Volume 2: Program Methodology

Ken Crum
Ron Dierker
Jackie Grody

NCI Information Systems, Inc.
3150 Presidential Drive
Building 4
Fairborn, OH 45324

Donald L. Thomas
Michael E. Clark
Maurice C. Azar
Steve Grace

Air Force Research Laboratory

July 2000

Final Report for the Period September 1995 to March 2000

Human Effectiveness Directorate
Deployment and Sustainment Division
Logistics Readiness Branch
2698 G Street
Wright-Patterson AFB OH 45433-7604

NOTICES

When US Government drawings, specifications or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from the Air Force Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161

Federal Government agencies registered with the Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
8725 John J. Kingman Rd., Ste 0944
Ft. Belvoir, VA 22060-6218

DISCLAIMER

This Technical Report is published as received and has not been edited by the Air Force Research Laboratory, Human Effectiveness Directorate.

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2002-0039

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Mark M. Hoffman

MARK M. HOFFMAN
Deputy Chief
Deployment and Sustainment Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 2000	Final - September 1995 - March 2000

4. TITLE AND SUBTITLE
Aircraft Battle Damage Assessment and Repair (ABDAR) Final Program Report, Volume 2: Program Methodology

5. FUNDING NUMBERS
C: F41624-95-C-5003
PE: 63106F
PR: 2745
TA: 02
WU: 05

6. AUTHOR(S)
Ken Crum, Ron Dierker, Jackie Grody, Donald L. Thomas, Michael E. Clark, Maurice C. Azar, Steve Grace

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NCI Information Systems, Inc.
3150 Presidential Drive
Building 4
Fairborn, OH 45324

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory, Human Effectiveness Directorate
Deployment and Sustainment Division
Air Force Materiel Command
Logistics Readiness Branch
Wright-Patterson AFB, OH 45433-7604

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-HE-WP-TR-2002-0039

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This is the second of three volumes describing the results of a program to develop technology to enhance the aircraft battle damage assessment and repair (ABDAR) process. This volume is intended for use by those individuals interested in developing and implementing an ABDAR system for operational use. The ABDAR Demonstration System has the demonstrated capability to enable the Combat Logistics Support Squadron (CLSS) assessor and technicians to significantly enhance the speed, accuracy, and completeness of the assessment process. This volume documents the program methodology employed to develop the ABDAR Demonstration System and demonstrate technology that provided the significant performance enhancements. After providing a program overview and overall approach, this volume details the methodology used to identify, create, prioritize, and document the ABDAR user requirements. It next provides the overall process used to design and develop the system that would meet the requirements in two different modes and handle two different types of electronic data. Software design is discussed including the software development environment, user interfaces, software and database architectures, and data development. Finally, the conclusions, lessons learned, and recommendations resulting from the program methodology are summarized.

14. SUBJECT TERMS
Aircraft Battle Damage Assessment and Repair (ABDAR) Content Data Model (CDM) Interactive Electronic Technical Manuals (IETMs) Computerized Fault Reporting System (CFRS) Integrated Maintenance Information System (IMIS)

15. NUMBER OF PAGES
84

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

THIS PAGE INTENTIONALLY LEFT BLANK

PREFACE

The research documented in this technical report was sponsored by the Air Force Research Laboratory, Deployment and Sustainment Division, Logistics Readiness Branch. This volume is the second of three volumes that summarize work performed to develop an Aircraft Battle Damage Assessment and Repair (ABDAR) technology to enhance the capability of Air Force technicians to assess damage, determine needed repairs and restore the aircraft to operational status. The work was funded under PE63106F, Project 2745. The work was performed under contract F41624-95-C-5003 by NCI Information Systems, Inc., with subcontractor support from Boeing Aircraft Company, RJO Enterprises, Inc., and GRACAR Corporation. Captain Michael Clark and 1st Lieutenant Steve Grace were the program managers for the major portion of the effort. Other Laboratory personnel who made major contributions earlier in the program were Captain Eric Carlson, Captain Floyd Gwartney, 1st Lieutenant J.C. Bradford, and 1st Lieutenant Maurice Azar.

This research could not have been accomplished without the support and assistance of many members of the Combat Logistics Support Squadrons, the Aircraft Battle Damage Repair Program Office, and the Air Force Materiel Command Logistics Directorate who served as members of the ABDAR Users Group, provided technical guidance throughout the program, and provided program advocacy.

The 653rd Combat Logistics Support Squadron, Robins AFB provided extraordinary support for the program. The 653rd provided the test facilities, test aircraft, and many of the technicians who participated in the field test. The squadron also provided the support of several of their instructors who served as subject matter experts and advisors throughout the program. The contributions of MSgt Ken McCain, TSgt Geoffrey Miller, TSgt George Boutwell, TSgt Ken Dockery, and TSgt Rob Meyers as technical advisors were invaluable and greatly appreciated by the ABDAR program staff.

The Program Methodology is the second volume of a three-volume final program report. It provides a description of the methods and techniques used to develop the ABDAR Demonstration System and a description of the system developed.

SUMMARY

The principal objective of this program was to develop and evaluate technology to significantly enhance the speed, accuracy, and completeness of the assessment of battle damaged aircraft. The approach adopted was to develop an automated capability to provide aircraft battle damage assessors with technical data and assessment tools via a portable maintenance aid (PMA). A demonstration system was developed and used to evaluate the Aircraft Battle Damage Assessment and Repair (ABDAR) concept. The ABDAR Demonstration System developed for the field test was an end-to-end system. It started with the aircraft debrief and continued through the ABDR process to final documentation of the damage assessment on an Air Force Technical Order (AFTO) Form 97. The system design was based upon a prioritized set of requirements identified by the ABDAR Users Group (AUG). The demonstration system provided the assessor with technical data for the testbed aircraft (F-15A), including applicable F-15 aircraft battle damage repair (ABDR) manuals and Technical Order (TO) 1-1H-39, Technical Manual General Aircraft Battle Damage Repair. The system supports two types of Electronic Technical Information (ETI). The first type of data, in the Indexed Portable Document Format (IPDF), presents technical data electronically in a format very similar to the paper TO. The second type of ETI, in the Content Data Model (CDM) format, provides technical data in an interactive mode. Sample technical data was developed in both formats. The benefits and effectiveness of the two formats was evaluated in a field test. The methods and processes used to develop the ABDAR Demonstration System are described in this volume of the final report.

A field test was conducted at Robins AFB to evaluate the effectiveness of the system. The evaluation was accomplished by having technicians assess simulated battle damage on an F-15 aircraft. Two thirds of the technicians assessed the damage using the demonstration system, and one third of the technicians performed the assessment while using the paper technical orders. Half of the technicians using electronic technical data used the CDM version, and half used the IPDF version. Three types of technicians performed the assessment task. They were fully qualified F-15 battle damage assessors, battle damage assessors qualified on another aircraft, and technicians (F-15 mechanics) who were not trained in aircraft battle damage assessment. Test results demonstrated significant benefits to using the ABDAR demonstration system for both the IPDF and CDM versions of the technical data.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

AIRCRAFT BATTLE DAMAGE ASSESSMENT AND REPAIR (ABDAR) FINAL PROGRAM REPORT

VOLUME 2: PROGRAM METHODOLOGY

The Aircraft Battle Damage Assessment and Repair (ABDAR) program final report consists of three volumes.

Volume 1, Executive Summary, contains a comprehensive summary of the objectives, methodology, results, conclusions, and recommendations of the entire program.

Volume 2, Program Methodology, contains an overview of the methodology used to accomplish the objectives of the ABDAR program.

Volume 3, Field Test, contains the results, conclusions, and recommendations, resulting from the field test.

The purpose of the present volume is to document the methodology used to produce the ABDAR Demonstration System, including requirements analysis, design and development of the demonstration system, and integration of the system with electronic technical data and other tools. It also briefly describes the use of the ABDAR Demonstration System in a controlled field test environment, summarizes the conclusions of the project, and provides recommendations for implementation of an ABDAR System.

INTRODUCTION

The goal of the ABDAR program was to develop and demonstrate technology that would provide a significant enhancement in the capability of USAF ABDR assessors and technicians to rapidly assess battle damaged aircraft. These individuals face the critical task of assessing, repairing, and returning battle-damaged aircraft to mission readiness during wartime. The ABDAR program built upon the concepts and technology developed in the Laboratory's Integrated Maintenance Information System (IMIS) program (Ward, et al. 1995, Volumes 1, 2, 3, and Thomas, 1995). The basic approach in the IMIS was to provide technicians with a portable maintenance aid (PMA) capable of presenting all technical and diagnostic information required to perform their jobs. A similar approach was adopted for the ABDAR program. Technology and a demonstration system were developed to provide the ABDAR assessor with information and planning tools needed to perform and document the assessment task. An ABDAR demonstration system was developed to evaluate the technology and evaluate its benefits.

The ABDAR Demonstration System developed in this project was an end-to-end system that supported the complete ABDAR process. It started at aircraft debrief and continued through the ABDR process to final documentation of the damage assessment on an Air Force Technical Order (AFTO) Form 97. The ABDR process and

requirements were supported with technical data from applicable aircraft battle damage repair (ABDR) manuals, including Technical Order (TO) 1-1H-39, Technical Manual - General - Aircraft Battle Damage Repair and specialized battle damage assessment manuals for a range of aircraft systems. The system handled two types of Electronic Technical Information (ETI) formats, Indexed Portable Document Format (IPDF) and Content Data Model (CDM) format.

Background

The ABDAR program was an advanced [6.3] research and development (R&D) project under the sponsorship of Air Force Research, Laboratory/Deployment and Sustainment Division Logistics Readiness Branch (AFRL/HESR). AFRL/HESR, along with the USAF and Department of Defense (DoD) ABDR communities, have long recognized that enhancements of this capability are critical to success in future armed conflicts. An enhanced ABDR assessment capability will provide an effective force multiplier to the Combat Air Forces (CAF).

The 55-month ABDAR program defined and implemented a concept developed in the early 1990's by AFRL/HESR. A preliminary demonstration of the concept was developed and demonstrated in 1994. The preliminary demonstration effort focused on devising a process to enhance the assessment and repair capability within the IMIS. The preliminary demonstration was intended as a module that would be tailored specifically for the assessor's use in an IMIS environment. A precept of IMIS was the integration of multiple sources of maintenance information, and this remained a common thread within the ABDAR program. The development challenge was to provide that information through a common user interface that operates on a workstation or PMA.

The preliminary demonstration of the ABDAR concept was conducted at an ABDR "Live-Fire" Demonstration Exercise conducted at Davis-Monthan AFB, AZ in October-November 1994. During this extensive series of ABDR activities, AFRL's preliminary ABDAR software, which emphasized an effective human-computer interface in the assessment of battle damaged aircraft, was demonstrated. The AF and DoD ABDR User Communities were very receptive to the AFRL concept and unanimously supported further development of the approach.

That fundamental early ABDAR research evolved into the current technology development effort. The basic approach taken was to perform a requirements analysis that would feed "As-Is" and "To-Be" modeling data and system requirements into the design, development, data authoring, integration, and testing for an ABDAR Demonstration System. Those processes began in August 1995 and culminated with the development of the ABDAR Demonstration System, a field-test to evaluate the system and final documentation in 1999-2000. Throughout the program, USAF and DoD users from the ABDR community were actively involved in the development of the ABDAR Demonstration System.

The ABDAR Demonstration System focused on the "assessment" portion of the ABDR process. The ABDAR Demonstration System demonstrates the technology to support multiple levels of ETI including Level II (IPDF type data) and Level IV (CDM type data).

ABDAR PROGRAM OVERVIEW AND APPROACH

The overall objective of the ABDAR program was to significantly enhance the speed, accuracy, and completeness of the assessment of battle damaged aircraft. Supporting objectives of the ABDAR program were to:

a. Provide generic assessment logic to support multiple levels of ETI.

b. Provide assessment logic that will complement any weapon system's ETI.

c. Employ new technologies, where appropriate, to aid in the assessment process.

d. Provide computer implementation software adequate to provide proof-of-concept of the assessment logic for two or more levels of ETI.

e. Provide the ability to operate successfully in environments having varying levels of connectivity (i.e., PMA-alone, PMA-ABDAR Server).

f. Provide transitionable ABDR performance enhancement technology to support maintenance operations.

g. Prove the benefits of the ABDR enhancements through a field test and/or demonstration.

The ABDAR program was conducted in three phases:

Requirements Analysis. The main objectives of the requirements analysis phase were to:

a. Identify and analyze the functional, information, and human-computer interface requirements for an ABDAR Demonstration System for operation in a combat maintenance environment.

b. Develop a system architecture, which supports those requirements.

c. Develop a system functional requirements specification.

d. Develop a System/Subsystem Specification (SSS). The SSS was the primary product of the requirements analysis phase.

Demonstration System Design and Development. During system design and development, a subset of ABDAR requirements was selected for implementation and

demonstration. Hardware and software were acquired, developed and integrated to implement these capabilities in the demonstration system.

Demonstration System Implementation and Field Test. Upon completion of the ABDAR Demonstration System, hardware and software were installed and field-tested at Robins Air Force Base (AFB), GA. Objectives of the field test were to:

a. Test the ABDAR concept under realistic operational conditions.

b. Evaluate effectiveness of the ABDAR Demonstration System in supporting the aircraft battle damage repair mission.

c. Demonstrate the technical advantages of ABDAR Demonstration System over the current Aircraft Battle Damage Repair (ABDR) paper-based method.

d. Identify strengths and weaknesses of the demonstration system for use in refining requirements for a production implementation of an ABDAR system.

NCI used an iterative prototyping method to develop the system. The major artifact in the ABDAR prototyping methodology is an Interim Software Demonstration (ISD). Six ISDs were accomplished. The ISDs were reviewed at the ABDAR Users Group (AUG) meetings where user feedback was obtained for the next iteration of system development.

By using the team approach and performing as a cohesive unit, the ABDAR program achieved an integrated system with a high degree of user acceptance. The ABDAR team obtained user acceptance by hosting AUG meetings that were effective in addressing requirements and acquiring feedback and approval in the following areas:

a. Requirements finalization and prioritization.

b. "As-Is" and "To-Be" model verification.

c. System requirements, process requirements, and user interface accommodations for an ABDAR Demonstration System.

d. ISD planning and reviewing.

The synergy that resulted from the ABDAR team and the AUG members interaction contributed significantly to the success of the final ABDAR Demonstration System.

Figure 1 provides a high-level schedule for each phase, including significant milestones.

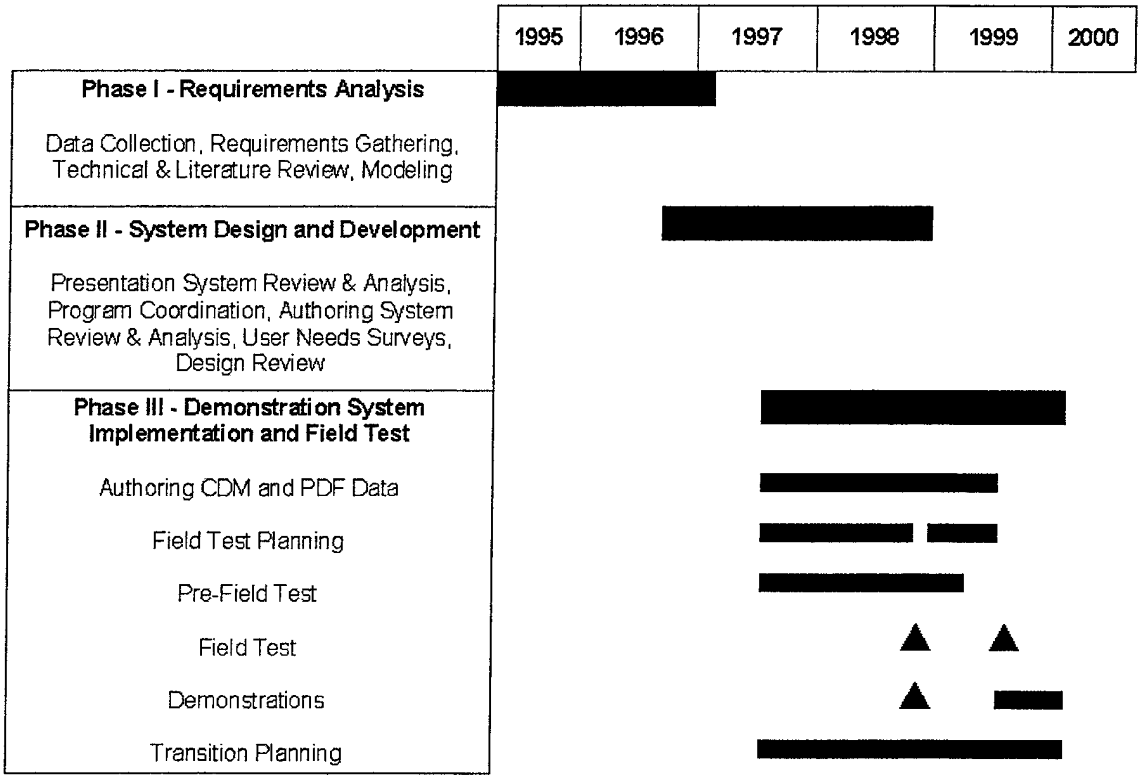


Figure 1 – High-Level Schedule

REQUIREMENTS ANALYSIS

A comprehensive requirements analysis was performed to identify, record, and track system requirements. The requirements analysis focused upon the needs of the end user, the aircraft battle damage assessor.

Data Collection

Methods used to gather and identify requirements for an ABDAR Demonstration System, included interviews with ABDR personnel, literature review, and a "user needs survey".

a. Interviews with ABDR personnel.

Over the course of several months, an NCI/government team visited 13 USAF bases. The trip number, location, site significance, and number of interviews that occurred at each location are provided in Table 1 – Data Collection Site Summary.

On the first three data collection trips (WPAFB, OH; Hill AFB, UT; and McClellan AFB, CA), interviews were conducted using two data collectors. One data collector was the interviewer and one was the data recorder. The interviewer's role was primarily to elicit the required information from the subject and control the pace of the interview. The data

recorder was responsible for extracting and recording pertinent details on the data collection sheets. After the third trip, the data collection team changed to one individual conducting the interviews. By this time, the interview process was perfected to the point that the single interviewer conducted the interview while also taking high-level notes. All interviews were recorded on audiotape, providing convenient reference in areas where the interviewer's notes were not clear.

Table 1 – Data Collection Site Summary

Data Collection Trip #	Location	Command	Site Significance	# Interviewed
1	Wright-Patterson AFB, OH	AFMC	445 CLSS F-16 Reserve Unit and Practice Interview Sessions	10
2	Hill AFB, UT	AFMC AFRES	649 CLSS F-16 Unit and Depot Engineers 419 CLSS F-16 Reserve Unit	29
3	McClellan AFB, CA	AFMC AFRES AFMC	652 CLSS A-10, F-111, F-117 Unit and Depot Engineers 604 CLSS A-10, F-111, F-117 Reserve Unit ABDR Program Management Office	22
4	Hurlburt Field, FL	AFSOC	H-53, H-60, MC/AC-130 ABDR personnel AFSOC performs its own ABDR internally	7
5	Tinker AFB, OK	AFMC AFRES	654 CLSS B-1, B-2, KC-10 Unit and Depot Engineers 507 CLSS B-1, B-2, KC-135, KC-10 Reserve Unit	20
6	Kelly AFB, TX	AFMC AFRES	651 CLSS C-17, C-5 Transport Unit and Depot Engineers 433 CLSS C-17, C-5 Transport Reserve Unit	10
7	Robins AFB, GA	AFMC AFRES	653 CLSS F-15, C-130, C-141 Unit and Depot Engineers 622 CLSS F-15, C-130, C-141 Reserve Unit	22
8	Moody AFB, GA	ACC	F-16, A-10, C-130 Composite Wing	3
9	Davis-Monthan AFB, AZ	ACC	Aerospace Maintenance and Regeneration Center, A-10	10
10	Charleston AFB, SC	AMC	C-17, C-141	5
11	Spangdahlem AB, GE	USAFE	F-15, F-16, A-10	15
12	Aviano AB, IT	USAFE	Deployed Scenario	6
13	Whiteman AFB, MO	ACC	B-2, Low-Observable (LO) Technology	11

Literature Review

A literature review was performed to avoid duplication of effort with other existing ABDR efforts and to identify other potential sources for ABDAR Demonstration System requirements. The review encompassed two actions: review documentation associated with the IMIS system, and review documentation associated with aircraft battle damage assessment or repair.

IMIS Technology Review

(a) The IMIS Final Report, consisting of three volumes (Ward, 1995, Volumes 1, 2, and 3), was reviewed for "lessons learned" information from the development of the IMIS demonstration system in the areas of hardware, software, communications, and organizational maintenance operations.

(b) The IMIS SSS was reviewed. The IMIS SSS (GDE Systems, 1995) included requirements for the ABDAR function, including information transfer, and interface requirements.

(c) The Interactive Electronic Technical Manual (IETM) specifications, Military Standards MIL-M-87269 and MIL-M-87268, were reviewed to assess areas of possible enhancement more efficiently support data requirements for ABDAR.

<u>ABDAR Literature Review</u>

(a) NCI reviewed existing literature on related efforts impacting this program, and documented significant findings.

(b) NCI investigated alternative sources of ABDAR requirements. AFRL/HESR conducted a search of the Defense Technical Information Center (DTIC) database and identified 63 documents related to ABDR. After reviewing the abstracts, 18 documents were selected, ordered, and reviewed. NCI also identified, obtained, and reviewed five Air Force Institute of Technology (AFIT) Masters Theses' and four other documents, obtained through other sources. A review of the "models" and other information in AFHRL-TR-83-25 (Wilper, et al. 1983) were included in this task. Each document underwent review and analysis to determine if it contained needs or requirements for an ABDAR Demonstration System. The basic purpose of these reviews was to search for relevant information on:

(1) Assessor aids (graphics, algorithms, and procedures).

(2) Technician aids (graphics, diagnostic tools, and procedures).

(3) ABDR engineering analysis methodology.

(4) Problems encountered in the field.

(5) Lessons learned from ABDR exercises

(6) Analyses of ABDR problems.

(c) A Document Review Form was used to record the results of each document review. The Document Review Form contained the document title, summary of the contents of the document, rating, and a reviewer comment concerning the document.

<u>User Needs Survey</u>

The final step in the data collection process was to conduct a survey of ABDAR specialists. The survey was used to finalize collection of data on the "As-Is" ABDR process. A survey consisting of 159 items describing the capabilities needed to accomplish aircraft battle damage assessment process was created, based upon the interviews and literature review. The survey was sent to 11 active and reserve Combat Logistics Support Squadrons (CLSSs). A total of 133 completed surveys were returned. The survey was used to:

1. Ensure correct interpretation of the data collected from the field interviews.

2. Provide an opportunity for users to indicate whether the item pertains to their job, and if so, how critical the item is, and how often they use the item to do their job.

3. Determine whether the needs list was complete and, if not, obtain the missing data.

4. Provide a tool to check the findings for accuracy, criticality, frequency of use, and completeness.

The survey items were listed in the form of need statements, many of which had multiple parts. For example, ABDR personnel were asked to rate the need for access to information on wiring. If they indicated that wiring information was needed to perform their job, they were then asked to rank the types of wiring information needed, this was provided on a subsequent list. A statement asking the respondent to fill in missing items of information followed each list. The ranking each item received varied depending on the respondent's job type. In this example, wiring information was much more critical to an Avionics Technician than to a Sheet Metal Technician.

The maintenance personnel rated the importance of each item using a modified Cooper-Harper scale. The scale was a 10-point scale, modified to be more compatible with the subject matter, using a tree structure to guide users to the appropriate rating. The rating was to be based on the respondents' level of need for the item to perform their assigned tasks.

Integrated Computer Aided Manufacturing (ICAM) Definition Methodology (IDEF)

Figure 2 illustrates the methodology for development of the IDEF models. The Architecture (IMISA) [General Dynamics Electronics Division (1990)] philosophy was the basic premise for the foundation of the ABDR models. The IMISA modeled the organizational-level maintenance world using IDEF methodologies. The ABDR environment was essentially an extension of the IMIS world. Duplication of effort was avoided by adopting relevant process models developed for IMIS. Combining the IMISA information with data collected in the field working with ABDR subject matter experts (SMEs) resulted in a realistic portrayal of the ABDR world and its interfaces to the O-level maintenance world modeled by IMIS.

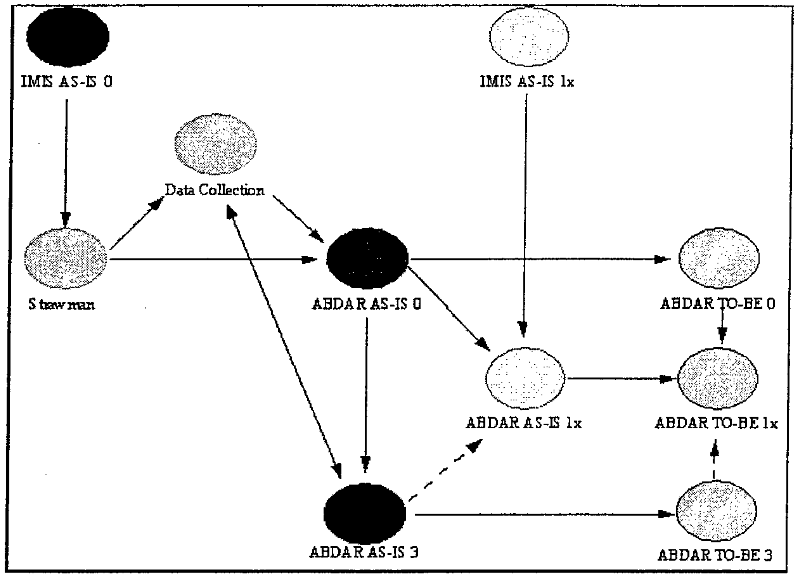


Figure 2 – Modeling Methodology

NCI used Knowledge-Based Systems, Inc.'s modeling tools to model the functions, processes, and information requirements of the ABDR domain. These automated tools enforced the IDEF methodologies and provided a means of analyzing the "As-Is" and "To-Be" models of the ABDR environment.

<u>IDEF0</u>. The IDEF0 model depicted a hierarchical representation of the activities and information flows in the ABDR world. Each activity identified had an associated text description, which described what was happening within that activity.

The IDEF0 "As-Is" model was the foundation on which the IDEF3 and IDEF1x models were constructed. Therefore, an extensive amount of preliminary work was expended in developing the IDEF0 model. Once this model was well established, work began on the IDEF3 Process Flow Network (PFN) and IDEF1x models. The IDEF0 model did not prove directly beneficial to the ABDAR Demonstration System development in that it would have been easier to develop the IDEF3 and IDEF1x models directly from the interviews and literature review.

<u>IDEF3</u>. The IDEF3 model was a PFN. This process-description-capture methodology did not incorporate information such as process times, cost, or other measurable data. In developing the PFNs, the lowest level activities of the function model (IDEF0) were extracted and ordered in a sequential fashion to develop the initial IDEF3 straw man. The activities were linked via relationships and junctions to describe the nature of the flow of events through the process. Some processes were sequential and some were concurrent.

Objects, facts, and constraints were identified for each of the activities in the PFN. Descriptions of the activities were extracted from the IDEF0 model, where appropriate. When descriptions were not readily available from the IDEF0 model (for activities that

were unique to the PFNs), they were created and reviewed by in-house SMEs. The descriptions were later reviewed by USAF SMEs during subsequent data gathering efforts. PFNs were used as use-cases (a more conventional software design artifact) in designing and coding object behaviors during system development.

IDEF1x. The ABDAR IDEF1x models were based on the IMIS "As-Is" IDEF1x model and the ABDAR IDEF0 model. The general process in developing the ABDAR IDEF1x model was to make a detailed review of the validated ABDAR IDEF0 model to identify the possible entities and attributes associated with the ABDR world. Once the attributes and entities were identified, an initial ABDAR IDEF1x model was developed. The next step was to review the ABDAR model regarding the interfaces to the O-Level world as modeled in the IMIS Architecture. Each entity and relationship in the IMIS "As-Is" IDEF1x model was examined for inclusion in the ABDAR IDEF1x model. Finally, the two groups were assembled into a coherent information model of the ABDAR world. IDEF1x was useful in defining database structures and conceptual models during system development.

System/Subsystem Specification (SSS)

The ABDAR SSS documented the requirements for an ABDAR Demonstration System. The ABDAR SSS described the system as encompassing activities, processes, and information associated with technicians, assessors, team chiefs, and maintenance supervisors. The specification is compatible with the System/Segment Specification for IMIS, which defined requirements for a maintenance support system at the base/wing level.

Identification of Demonstration Requirements

The requirements in the ABDAR SSS prescribed the behavior of an operational implementation. The most important of these requirements were selected for implementation in the ABDAR Demonstration System for evaluation. Requirements were selected based upon the requirements analysis, and were reviewed by the AUG members.

The following assumptions were developed from the ABDAR program SOW and preliminary plans for the field test and demonstrations. These assumptions were used in designing the demonstration system to ensure that the system would meet field test requirements. For the field test, the ABDAR Demonstration System must be compatible with the following conditions/constraints:

a. Users

   1. One or more assessors.

   2. One or more technicians.

3. One team chief.

4. One or more engineers.

5. One manager (plays role of outside world, production super, supply, explosive ordnance disposal [EOD], etc.).

6. Each user will be able to use his/her own PMA.

b. External Communications

1. The ABDAR Demonstration System will simulate a connection with the Air Force's Integrated Maintenance Data System (IMDS). A data collector will input any necessary responses from the IMDS environment including.

(a) Provide a production superintendent status board.

(b) Provide resource information.

2. Self contained Intranet (no Internet communications).

3. Will provide connection to the Computerized Fault Reporting System (CFRS).

c. Forms

1. AFTO Forms 781A and 781K.

2. AFTO Forms 97 and 97a.

3. AF Form 2005.

d. Computer Environment

1. Pointing device and full keyboard.

2. Non-removable storage medium.

3. At least one removable storage medium.

4. Personal Computer Memory Card International Association (PCMCIA), parallel and serial ports.

5. Windows New Technology (NT).

6. Wireless packet based network capability.

7. Super Video Graphics Array (VGA) Color.

8. A server and a client will exist as separate pieces of hardware connected through a network interface.

e. Test Environment

1. One damaged F-15A Aircraft.

(a) Types of damages were System, Structure, and Wiring.

(b) Locations of damages were Door 6R and Left Wing Trailing Edge.

2. One ABDAR Tool and Material Kit.

f. Inputs and Outputs. The inputs to the system will be those initiated by the user(s) through the graphical user interface (GUI), by the data collector simulated system status boards, and by a download of debrief information from CFRS. Outputs of the system will be:

1. Communications messages to the manager requesting assignment of resources and approval of assessor recommendations.

2. Communications messages from the manager responding to requests for assignment of resources and approval of assessor recommendations.

3. Documentation forms at the completion of the demonstration scenario.

The above assumptions are stated without modification as they were generated prior to development of the ABDAR Demonstration System. Clarification of these assumptions was required during system development.

Data Requirements

When development of the ABDAR Demonstration System began, there were two predominately electronic data types (PDF and CDM) available to weapon system Special Program Offices (SPOs). The Joint Computer-Aided Logistics Support (JCALS) program office was in the process of converting all paper documents into PDF format, while the newer weapon systems were producing TOs in CDM format. Because these two formats vary significantly, in how they are stored and displayed, it was conjectured that a system that could make effective use of either format would have significant advantages. Consequently a requirement was added that the ABDAR Demonstration System must be capable of presenting both CDM and PDF based TOs for ABDR maintenance.

PDF data is (or can be made) available for all current weapons systems. In its simplest form, PDF data is nothing more than scanned TO pages, which are viewed on a computer using Adobe Acrobat. Each TO is typically stored in a separate flat file. To aid in navigation, hyperlinks can be added to a PDF file using Infolinker, an Adobe Acrobat plug-in. A PDF file with hyperlinks is commonly referred to as an Indexed

Portable Document Format (IPDF) file. JCALS compliant IPDF files contain, at minimum:

a. A hyperlink from each Table of Contents entry to the page on which that entry can be found.

b. A hyperlink from each reference within the text of the file that points to a paragraph, table, or figure within the same IPDF file.

There is little technical value of IPDF over paper from an automatic assessment or diagnostic tool perspective. The value of IPDF lies in the ability to have most (if not all) technical data available at the aircraft, to display visible links, and to perform keyword searches within a single IPDF file or within an entire TO library. For the ABDAR Demonstration System, hyperlinks to external IPDF files were also generated so the users had an entire TO library at their disposal. This eliminated the need for users to manually open additional IPDF files when a reference was encountered.

Some current and most weapon systems currently under development use CDM data. The data is stored in a hierarchical discrete data format that allows for parsing of information in discrete parts. This hierarchical discrete data format allows for a high degree of interactivity and dynamic diagnostics. A typical system using CDM data has the ability to ask the user questions, perform calculations, and store autonomous pieces of data.

CDM data authoring currently requires extensive human intervention. The authoring is analogous to completely redeveloping data that has already been developed for a paper environment. This process is both time consuming and expensive. Under current budget conditions, it is unlikely that CDM data will be developed for existing weapons systems. A CDM data solution becomes much more cost effective when the Technical Data is generated directly from the electronic information used to design a new aircraft, as is currently being developed for the F-22 and CV-22.

Technical Requirements

One of the major problems with exploratory software projects is they can become obsolete before they are released. To help mitigate this issue, the ABDAR Demonstration System was developed with emerging technologies in mind. Paralleling the development strategies and technologies currently being introduced into the industry sector provided an effective way to ensure that the ABDAR Demonstration System architecture would be current when delivered to the Government. The predominate paradigm driving most Information Technology (IT) development is the World Wide Web (WWW). Therefore, it was prudent to develop the ABDAR Demonstration System using the same technologies.

Since the WWW was introduced in 1991 by the CERN laboratory, its growth has been astronomical. The business community is discovering that it can deliver content specific data to users using this relatively new technology. The Web is beginning to

compete not only in the mass media market place, but also in markets typically reserved for Information Systems (IS). Currently users can view video clips, sound bytes, and 3-D images describing just about anything regarding a product, and then through form based interactions, order that product immediately. Additionally, with the new communication tools available through the Web, a customer can send electronic mail or use audio or videophones to directly communicate with the manufacturer of the product if the customer needs any additional information. This entire scenario parallels the requirements of any IMIS or ABDAR System. A technician on the flightline needs to find some information about a weapon system, order parts for that weapon system, and communicate with someone if that information is incomplete. The ABDAR Demonstration System attempted to leverage research, being performed in the commercial world, on content specific data.

An additional search was performed to find other technologies or applications that could be used to aid the assessor in performing ABDR. The SOW requirement to produce a single integrated system to support the entire ABDR process precluded the use of most applications. Two applications, Wiring Illuminator (WI) and Computerized Fault Reporting System (CFRS), were identified as having the potential to greatly assist the ABDR process and were included as requirements of the ABDAR Demonstration System. WI increased the speed of wiring assessment and repairs by displaying relationships between electrical wires and aircraft systems. With some modification (using wire bundles rather than individual wires), the WI tool was designated for implementation as part of the ABDAR Demonstration System. CFRS was selected to support the debriefing portion of ABDR. Since much of the actual functionality of CFRS is outside the scope of ABDR maintenance, a simulated output was to be directly uploaded into the ABDAR Demonstration System database for use in the field test.

This section, Requirements Analysis, covered tasks accomplished during the first 18 months of the program. Data collection consisted of base visits, literature reviews, and user needs surveys. IDEF methodology (IDEF0, IDEF3, and IDEF1x) modeled the "As-Is" and "To-Be" worlds of ABDR. The ABDAR SSS documented the requirements for an ABDAR Demonstration System. The ABDAR SSS described the system as encompassing activities, processes, and information associated with technicians, assessors, team chiefs, and maintenance supervisors. Identification of demonstration requirements concluded this section of the report by listing the assumptions developed from the ABDAR program SOW and the preliminary plans for the field test and demonstrations using the ABDAR Demonstration System. Additionally, the Assessment Logic was defined during the requirements analysis phase, and refined throughout the ABDAR program through inputs from SMEs and AUG members. The next section highlights the process the assessor uses while performing ABDR maintenance.

ASSESSMENT LOGIC

Research of the logic used in the assessment of battle damaged aircraft was accomplished during the Requirements Analysis portion of the ABDAR program. Development of a common understanding and definition of the term assessment logic

was the initial consideration. The following statement was presented to the AUG #4 meeting conducted in January 1998.

> "Assessment logic is the thought processes that the human should apply when faced with real time information (clues) which needs to be processed (decide what it means or what to do with it)."

After review, consideration, and discussion, that definition was expanded to provide the following formal definition used by this effort.

> "Assessment logic is defined as the deductive thought process that should be applied when evaluating or appraising the condition of battle damaged aircraft. Feeding the deductive process are the real time clues (information) used to infer the repairs needed to return the aircraft to a mission capable status."

Research identified many different examples of assessment logic used in the assessment and repair of a battle damaged aircraft. By grouping the examples, three general types of assessment logic become apparent. The three types are Cause and Effect, Sequencing, and Decision-Making.

a. Cause and Effect. Cause and effect assessment is focused upon determining the relationships between a "known condition and a future condition or a known condition and its root cause." During aircraft assessment, there are situations where the condition is known but the future impact is unknown. More specifically stated, the assessor identifies or suspects a problem with a damaged entity and needs to determine what affect that problem may have on the system or the aircraft mission worthiness. In these types of situations, the assessor is looking forward in time. Conversely, situations arise where a current condition is known and the root cause is unknown. For example, the assessor sees some sort of abnormal behavior and must determine which component is responsible for that behavior. These situations look backwards in time. To further complicate matters, what may be the cause in one situation may be the effect in another.

b. Sequencing. Many assessment situations that require logical ordering or sequencing occur. The "what to do next" questions are posed by assessors and technicians throughout the entire ABDR process. The deductive reasoning required for determining the next step is not always obvious. It may be as simple as determining what information needs to be collected or evaluated next, or as complicated as being faced with a triage approach to allocating resources to assess and repair multiple aircraft. When trying to answer the question of "what to do next," the assessor relies on techniques to make the problem more manageable (event sequencing, decomposition, and type distinction).

c. Decision-making. The deductive thought processes involved in the decision-making portions of the assessment process entail the gathering of essential bits of information, evaluating each as it interacts with the other, and reaching a logical

conclusion. The largest decision-making domain involves the evaluation of options against needed/available resources, including time, to produce a mission ready aircraft. The second largest domain is the planning of the actions and resources to accomplish the repairs.

Implementation of the general types of assessment logic into the ABDAR Demonstration System was dependent on the intelligence in the ETI. The portions of the ABDAR Demonstration System that are dependent upon intelligent ETI (i.e., CDM) are tightly coupled to the data format. Different implementation solutions are used in the ABDAR Demonstration System. It contains many different methods of sequencing, "Wizards" to assist in the decision-making processes needed to make accurate decisions and perform proper documentation. Also included is data to assist the user in determining the cause and effect relationships. Not all ABDR scenarios are typical however, so there is no single solution to implementing "Assessment Logic". Assessment Logic is dependent upon many factors including the amount of effort expended versus the advantage gained in a particular application. Detailed Assessment Logic results can be found in the separate Assessment Logic report (NCI Information Systems 1999).

ABDAR DEMONSTRATION SYSTEM DEVELOPMENT

The ABDAR Demonstration System was developed using a methodology typically called evolutionary prototyping. Evolutionary prototyping is a life-cycle model, which defines the system concept as the product is developed. The artifacts from the requirements analysis phase (Demonstration SSS, Assumptions, and IDEF Models) were used to define the development approach and to feed each development cycle. Development of the most visible aspects of the system was first. These visible aspects were demonstrated to the user through an ISD, and then the next ISD was developed based upon user feedback, system performance, and the additional requirements needed for a fully functioning system. Figure 3 illustrates the evolutionary prototyping life cycle.

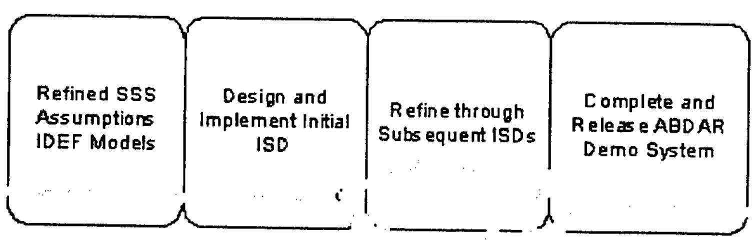


Figure 3 – Evolutionary Prototyping Life Cycle

This approach provided the flexibility required for addressing new technologies and any unforeseen changes within the USAF maintenance structure. To avoid developing a system that met the stated requirements but provided little utility, the ABDAR team, through periodic reviews by the AUG members, ensured the final ABDAR

Demonstration System was a tool that was useful. The evolutionary prototyping method produced steady, visible signs of progress and was especially useful given the strong demand for development speed.

NCI produced four ISDs on the ABDAR Demonstration System and one ISD that explored advanced concepts outside the realm of the field test. The final ISD (#6) documents the field-test version (ISD #4) of the software for delivery to AFRL/HESR.

Several processes were utilized in the development of each ISD. These processes consisted of a concept definition, code development, data development, AUG review, Human Factors Engineering Approach review, and analysis (see Figure 4). NCI maintained the software and data associated with each ISD development. Standard configuration control procedures were used to manage the data. NCI, upon meeting the acceptance criteria outlined in the Concept and Analysis papers, delivered each ISD to the Government.

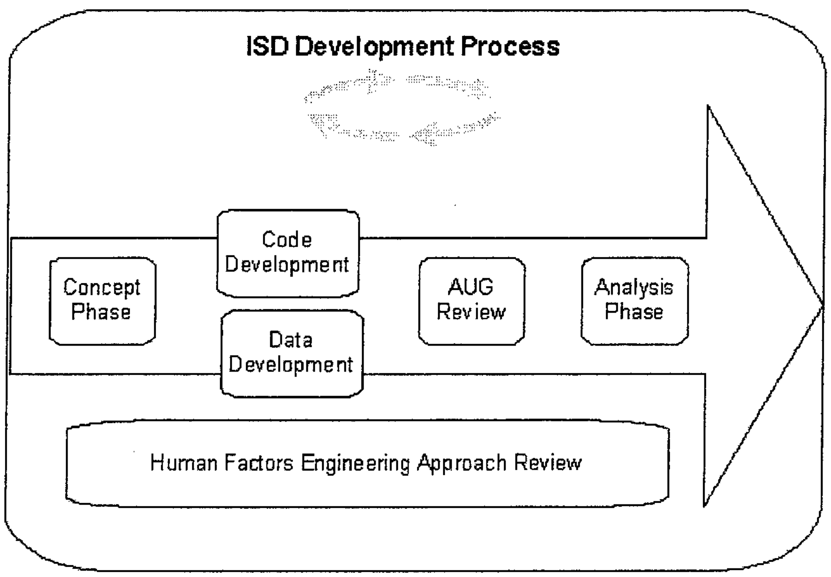


Figure 4 – ISD Development Process

a. Concept Definition. Preceding each ISD, NCI documented the objectives and approach of the ISD in a concept paper. The objectives and approaches for different ISDs varied in nature. Some ISDs were exploratory in nature, implementing and testing a new technology, configuration, or specific tool. Other ISDs were more evolutionary, addressing specific requests from user responses during the AUG meetings or implementing refined requirements identified from the SSS. Each concept paper outlined the expected system requirements necessary to build and execute the ISD. A list of the refined requirements was addressed and included in each concept paper.

b. Code Development.

1. The code development approach for ISD development was based upon the Booch Methodology, but was flexible enough to handle the unique requirements for each ISD. The Booch Methodology is an object-oriented approach that uses classes

and objects as its conceptual framework. It supports the four major elements of the standard object model:

(a) Abstraction
(b) Encapsulation
(c) Modularity
(d) Hierarchy

2. The Rational Rose modeling tool was used to develop the use-case and conceptual models for the ABDAR Demonstration System. ERwin was used to create data models and views necessary to support the assessment objects. The object-relational link between these two models allowed NCI to maintain compliance between the database and the objects or components. Figure 5 - ABDAR Development Process, illustrates this relationship.

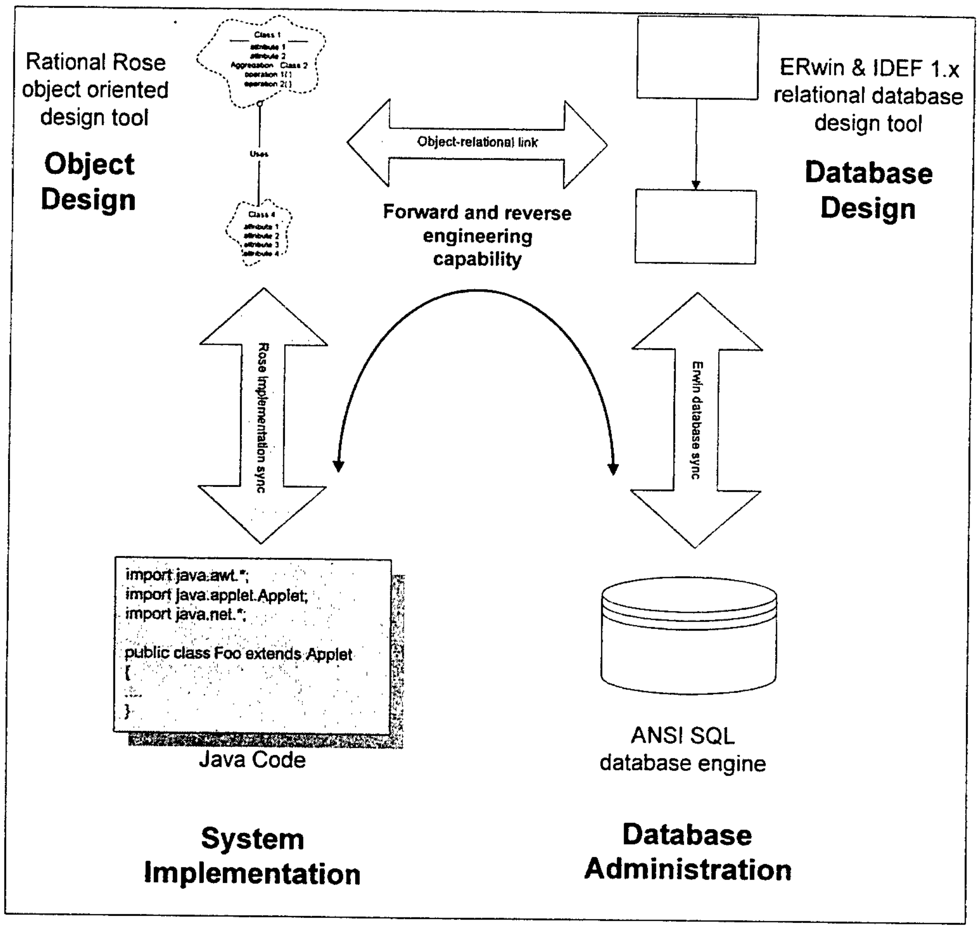


Figure 5 - ABDAR Development Process

3. Upon completion of the ISD specific models, code development responsibilities were determined. A vertical approach to development was usually chosen, with various individuals being responsible for a horizontal layer (client, application layer, database layer, and external components). For example, if a vertical function such as Detailed Inspection is selected to be completed, one programmer is

responsible for the Detailed Inspection Page; one is responsible for deploying the components necessary in the application layer; another is responsible for developing the database schema; and yet another is responsible for developing the viewers. An internal completion date was determined, and when all the layers were sufficiently tested, the software was internally released. If the vertical development necessitated a release of ETI data or WI, a programmer was assigned responsibility for coordinating the effort.

4. Upon acceptance of each ISD by government program manager, the ISD was configured and maintained at NCI. The ABDAR team had access to the ISDs through a test environment maintained at NCI. NCI conducted quality assurance activities to include discrepancy and risk tracking, where appropriate, during the development of each ISD. The software was maintained through internal configuration control during each ISD's development.

c. Data Development.

1. Throughout the ISD development process, NCI and Boeing produced data to support the testing of the current ISD, while concurrently developing data for the field test. NCI was required to develop a CDM viewer to support the ABDAR Demonstration System. The viewer was developed incrementally with the ISDs. Therefore, the development of each ISD [on the ABDAR Demonstration System] depended upon having sample CDM data sets available for the software developers to access and manipulate. To fulfill this requirement, NCI created test data sets using an authoring environment identical to the one used by Boeing. This allowed the developers to access a small data set, evaluate ways to accomplish the logic function, and give feedback to the authors to either continue in the same manner or change the authoring approach. The test data sets were small and easily changed to try new implementations. Results from this were cycled back to the Boeing authors to ensure the data authored fulfilled the requirements designed into the ABDAR Demonstration System. Since NCI used Adobe Acrobat Exchange to display IPDF data, incremental release of this data was not required.

2. As versions of the field test data became available, they were tested with the appropriate ISD. Boeing was responsible for the initial development of the ETI along with data needed for the WI operation and data needed to support debrief by CFRS. NCI was responsible for the final development as well as system data to support resource management, and aircraft status data. NCI was also responsible for enhancement, integration, and testing of the ETI after Boeing completed the initial development.

d. AUG Review. During the development process, the ABDAR team made extensive use of the AUG. Members of the group were selected from CLSS organizations, ABDR personnel from other operational units, the ABDR Program management Office (PMO), contractor personnel, and other knowledgeable individuals who were willing and available to participate. The purpose of the group was to ensure

all user requirements were incorporated into the ABDAR Demonstration System and to collect periodic feedback concerning to the usability of the system.

e. Human Factors Engineering Approach Review. This review was accomplished throughout the process of each ISD development.

1. Applying contemporary human factors principles to the design of a Website was a challenge. At the outset of the ABDAR program, there were no standards for developing Websites to be consistent with established human factors principles. For this reason, the utilization of Windows standards was consistent with the goal of establishing a sound design for a Website. Issues specific to the WWW and its browsers were addressed according to browser conventions.

2. The purpose of the ABDAR Demonstration System was to improve the speed, accuracy, and completeness of the ABDR process. To accomplish these goals, the ABDAR Demonstration System provided tools to improve the information processing and decision-making capabilities of the ABDR personnel in stressful, real-world situations. Four critical issues were addressed in the user interface design, early focus on users, interactive design, empirical measurement, and iterative design.

(a) *Early focus on the user* - During the AUG meetings, NCI established an early focus on the user. The user guided the interface design effort. Early focus also included developing the interface first, then working the overall system design. As documented in the concept and analysis papers, NCI presented a set of prototype interfaces dubbed ISD #1 through ISD #4 to the AUG members.

(b) *Interactive design* – This included using the special knowledge of SMEs. The collective knowledge of the SMEs proved invaluable in the design effort.

(c) *Empirical measurement* - Throughout the design process, AUG members completed quantitative evaluation forms rating the ABDAR Demonstration System for usability and ease of learning.

(d) *Iterative design* – The iterative software development design allowed us to collect empirical data during the AUG meetings to be analyzed for subsequent ISDs. This principle was inherent in the ISD concept.

3. Adherence to these guidelines assured that the software design process produced a human-computer interface that was user-friendly, learnable, and consistent with human factors design considerations.

f. Analysis Phase. Upon completion of an ISD, including AUG review, NCI documented the results of the ISD in an analysis paper. The analysis paper for each ISD contained user evaluations, if any, or analyses of system performance. It documented the level to which each of the refined requirements was addressed. The analysis paper documented which of the refined requirements required modification due to new user input. An important phase of evolutionary prototyping was the documentation of the impact and accomplishments of the ISD in the life cycle of the

project. All development models and software code fell under configuration control and a baseline was achieved.

ISD #1- HyperText Markup Language (HTML) Prototype

The objective of ISD #1 was to present visual objects to the AUG members allowing their input to guide the definition of the functionality and appearance of the ABDAR Demonstration System. HTML was the predominate format used to compose the visual objects for ISD #1. HTML is the set of "markup" symbols or codes inserted in a file intended for display on a WWW browser. By using HTML for the first ISD, the users could be introduced to the concept of an ABDAR Demonstration System using browser technology. Additionally, HTML allowed the developers to quickly generate and modify visual objects. Unfortunately, the functionality necessary to achieve the requirements of the ABDAR Demonstration System was not provided by HTML and was therefore replaced by Java components in subsequent ISDs.

The AUG members input assisted NCI in defining the basic system behavior and in exploring concepts of how the ABDAR Demonstration System was to be utilized in the field. The information collected from the AUG members, on ISD #1, was used to guide the design of the objects that made up assessment logic and the necessary database views to support those objects. ISD #1 was also utilized to identify containers to hold the visual objects, identify commonality among the visual objects, develop meaningful symbols to represent visual objects, and refine the behavior of the visual objects. The assumption was made that by incorporating the users' perspective early in the design process, the ABDAR team would be better able to ensure that the final ABDAR Demonstration System would serve the users' need.

ISD #2 – Damage Collection

The rationale for ISD #2 was to begin system development on the portions of the ABDAR Demonstration System that were considered the highest risks and that contained the highest potential payoff to the program. The most significant components that concentrated on the core assessment functions were Damage Collection and Repair Planning. Of these two, Damage Collection was the primary focus of ISD #2. Damage Collection was considered the higher risk because of the unique approach NCI used in implementing technical data presentation, a core piece of Damage Collection. NCI separated the presentation of the technical data from the rest of the system because of the requirement to display two technical data types. Separate components (viewers) were developed for each type. Each of these viewers, when coupled with a Damage Collection form made up the Damage Collection component. This approach had not previously been attempted in an IMIS-like system using CDM data.

Additionally, the most significant development risks were communication between components (specifically maintaining database transactions that span Web pages) and implementation of a system using an emerging technology (Java). Complete database transactions were just being introduced to the WWW environment. During the design of

ISD #2, most Web architectures were stateless in nature. An architecture known as Beans Connect was used to introduce state to the ABDAR Demonstration System.

To maintain user involvement, screen prototypes were generated for the entire system and shown to the AUG members. The prototypes used in conjunction with working components maintained a complete look and feel of the entire ABDAR Demonstration System. Additionally, the default user functionality that comes with a Web server was demonstrated to elicit user feedback.

ISD #3 – Repair Planning

Repair Planning was the primary focus for ISD #3. To maintain a smooth transition from Damage Collection to Repair Planning, the Repair Selection component was also developed. ISD #2 components were migrated to Swing components in ISD #3. Other implementations in ISD #3 were the incorporation of the WI and Computer Graphics Metafile (CGM) viewer into the CDM version of Damage Collection. The IPDF version of Damage Collection was implemented with a component to control the Adobe Acrobat Exchange.

In ISD #2, communication between components and maintaining database transactions that span Web pages was solved using Netscape Beans Connect. By the beginning of the ISD #3 development, Netscape no longer supported Beans Connect. Therefore, a multi-tier architecture was developed for use in ISD #3 and impending ISD's. To this end, ISD #3 implemented a design based upon the Enterprise Java Bean (EJB) 0.8 specification by Sun Microsystems. The application server (or middle-tier) was populated with the problem-specific logic commonly called business objects.

A database architecture was deployed to support the remaining iterative prototypes of the software. The database was implemented to evolve along with the prototype applications. Three separate database instances, or regions, were developed for ISD #3. The first instance consisted of a "workspace" for database objects under construction containing very rudimentary data. The second instance contained "fairly" stable database objects for use by the developers generating code using the Java language, along with more realistic and frequently refreshed data. The third instance contained the final set of database objects and final test data delivered at the end of the ISD. All objects within an instance belonged to a group of common owners, depending upon the type of data contained in each object. Procedures were constructed for each instance so that the database could be reset to contain a pristine dataset for demonstration and testing purposes. Implementation of security measures allowed execution only by privileged users.

ISD #4 – Field Test Version

ISD #4 was a robust system that could support a field test, as outlined in the ABDAR Field Test Plan. At this stage, the biggest risks to the development of the ABDAR

Demonstration System were an immature development language and environment, and the requirement for ease of using the system by test subjects, with minimal training.

Maturation of most of the tools used to develop the basic system architecture had stabilized, minimizing several technical risks for producing the ABDAR Demonstration System. The ABDAR Demonstration System was migrated to these new standards, most notably, EJB 1.0 and Java 1.1. One risk, not minimized through product maturation, was the delivery of the ABDAR Demonstration System through an Internet browser. Integral to the original system design, including the browser, was the Java Plug-In component, developed by Sun Microsystems. The Java Plug-In was still too unstable and did not make consistent use of the Java 'garbage collection' feature, causing the client side of the application to use an inordinate amount of memory, resulting in a system freeze. To mitigate this risk, modification from a series of applets to a single application was made to the ABDAR Demonstration System. This modification eliminated the necessity to execute the ABDAR Demonstration System through an Internet browser.

The second risk, identified during drop testing of the ABDAR Demonstration System, was the amount of training required before an assessor could effectively use the system. Developing wizards and a tutorial, as part of ISD #4, mitigated this risk. The tutorial was a [hardcopy] step-by-step set of instructions for using the ABDAR Demonstration System. Definitions of system and domain specific terms were provided to aid the assessor during use of the system. Development of context-sensitive wizards aided the assessor on-line. In addition to simple instructions, these wizards performed some system specific tasks that were not obvious to the untrained user.

For the previous ISDs, a single database instance had been created. For ISD #4, there were two servers (a development server and a field test server), each containing two instances. At this point in the development, the multiple instances were maintained to support concurrent development and testing.

ISD #5 –ABDR Technology Concepts

Unlike previous ISDs, ISD #5 was not an iterative step in the development of the ABDAR Demonstration System; rather, it was a collection of thought-provoking ideas to enhance the ABDR assessment process. The ISD #5 concept and analysis paper documented the ideas and results produced during that effort. This paper is provided in the Appendix.

ISD #6 – Documenting the Field Test Version

NCI concentrated on documenting the field test version (ISD #4) of the software for delivery to AFRL/HESR at program end as ISD #6. No major software, hardware, or database changes were made.

This section, ABDAR Demonstration System Development, covered the incremental development approach used in the implementation of the ABDAR Demonstration System. It highlighted the substantial involvement of the users and the requirements analysis as key elements in this process. Additionally, a history of the ISDs was presented. The next section, Software Design, focuses on the artifacts from the software development as a result of the final field-tested version of the ABDAR Demonstration System (ISD #4).

SOFTWARE DESIGN

The primary focus of the software design for the ABDAR Demonstration System was in developing a design that was flexible enough to change with new technology and as user needs became more defined through the prototyping process.

Overview

The primary software constraint in developing an ABDAR System was the generation of a generic ABDR assessment tool that would operate independently of the data type used to store the technical order information. A secondary constraint was the necessity that a fielded ABDAR System work within an integrated maintenance environment with other information systems (most notably, IMDS). These two constraints led NCI to use a component-based approach for the design and development of the ABDAR Demonstration System. Additionally, the ABDAR Demonstration System was developed using advanced technologies whenever possible. Since the Internet is an emerging dispersed technology that supports distributed component architectures, an Internet metaphor was chosen as the overriding architecture.

Separation of Assessment and Technical Data

The speed and accuracy with which an aircraft can be returned to mission capable status is foremost to the mission of ABDR. Generally, this mission requires the identification of the most effective repair(s) for an aircraft. In identifying the repair(s), the assessor relies upon a variety and wide range of information. The data and information used varies from physical and functional diagnostics, to availability of resources and the status and mission of the aircraft. Through the dissemination and integration of the information, the assessor creates a dynamic repair plan for fixing the aircraft. The ABDAR Demonstration System aided the assessor by providing tools that assisted in identifying and integrating the proper data and information. The tools provided and maintained information about the state resources of the battle-damaged aircraft and information about the damages discovered on the aircraft.

The essence of the ABDAR Demonstration System was to provide tools to aid the assessment process. Both the tools and the assessor rely upon technical orders to properly support assessment of aircraft. The data format used to store and deliver

technical orders is not consistent for the different airframes. Consequently, the ABDAR Demonstration System was to demonstrate that these assessment tools could be executed regardless of data type. To demonstrate data independence two formats were chosen, CDM and IPDF, the ABDAR Demonstration System ran with either. Because assessors will be required to work on various airframes, with differing data types, it was prudent to make the tools work similarly, regardless of whether they were supported by CDM or IPDF. A viewer (Figure 6) was developed to display CDM data and Adobe Acrobat Exchange was used to display IPDF data. Interfaces were developed to support the use of both the CDM and Portable Document Format (PDF) viewers. Although presentation of the assessment tools to the user still varied based on the supporting data type, the design allowed developers to minimize the differences.

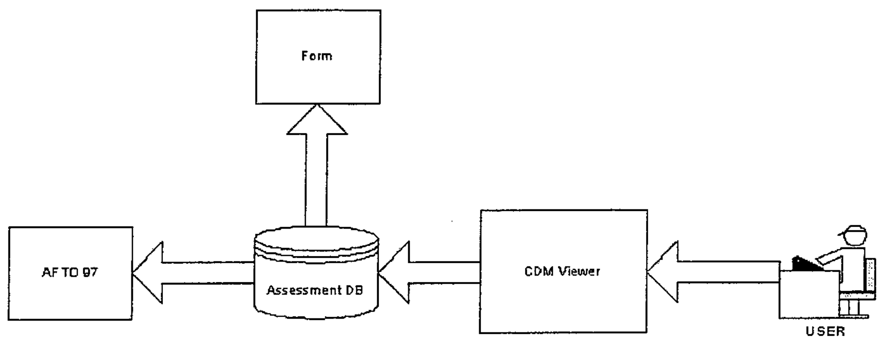


Figure 6 – CDM Viewer Implementation

a. CDM data, the first type of data, was compliant with MIL-STD-87269. The CDM data contained distinct, structured pieces of information known as data elements. The data elements were parsed from SGML files into a relational database. CDM data is often referred to as intelligent data. From the user's perspective, the data and its the viewer appear to do most of the assessment work. When the viewer component was opened, it presented the user with a visual prompt. This prompt could be a graphic from which the user selects a damaged region, or a more direct question requesting the size of a specific damage. After the user entered the requested information, the viewer sent the data to the ABDAR Demonstration System through an Application Program Interface (API). The data sent might have been as simple as a single data element, in which case the system simply recorded the data for final documentation and displayed it to the user. Alternatively, the data may have been as complex as a list of actions (repairs or evaluations) that needed to be performed. In either case, the user had minimal interaction with the assessment tools in the ABDAR Demonstration System, other than to verify the information produced by the CDM viewer. Most of the interaction was with the CDM viewer.

b. The second type of data was stored in IPDF format. Since IPDF does not contain identifiable distinct pieces of autonomous information that can easily be extracted for the population of a relational database, it was left in a flat file format. The Adobe Acrobat program, used to display the IPDF data, used the flat files stored locally on a PMA loaded with the ABDAR Demonstration System.

Because of the nature of IPDF, Adobe Acrobat cannot be as interactive as a CDM viewer. Additionally, there was little information that could be extracted from Adobe Acrobat that could directly populate the assessment tools. From the user's perspective, the PDF viewer and the assessment module were not directly linked (see Figure 7). Using Adobe Acrobat the user was required to navigate to the proper evaluation instructions in the IPDF TO and enter the information into the assessment forms. The data was presented on the display in a format very similar to the corresponding paper TO. In the IPDF version, the user was required to determine which evaluations and repairs needed to be performed to assess the aircraft. The user then had to create them using the ABDAR assessment tools. This is contrary to the CDM version, which accomplished these tasks for the user. The only information passed to the assessment module from the Adobe API was a bookmark containing the current TO and page.

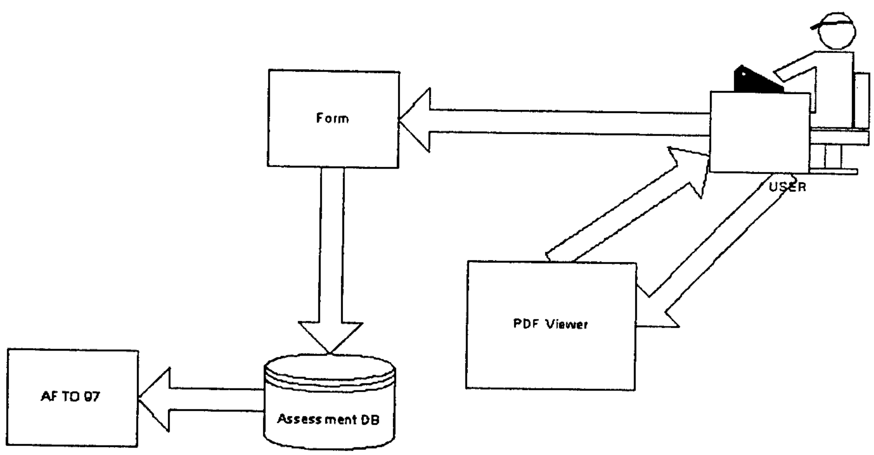


Figure 7 – IPDF Viewer Implementation

Component-Based Design

Components and containers were the two basic abstractions on which the component model was founded:

a. Components can range in size and capability from small graphical user interface (GUI) widgets like a button, to a more full sized application such as an HTML browser. Components can have visual appearance, such as a button, or can be non-visual, such as a data feed monitoring component.

b. Containers were used to hold an assembly or related components. Containers provided the context for components to be arranged and to interact with one another. Containers are sometimes referred to as forms, pages, frames, or shells. Containers can also be components (i.e., a container can be used as a component inside another container).

The distributed component design needed to implement the ABDAR Demonstration System fit well onto a Web-based architecture. The Web is based upon the classic

client/server or *n*-tier paradigm. Client/server, as applied to the Web, means that a computer running Web server software exists somewhere, and many different users, called clients, using Web browsers or specific client programs, can access containers and components from the Web server. The Web or Internet is a distributed view of this paradigm, with servers all across the globe. For the ABDAR Demonstration System, a full Internet was not required. An Intranet or Extranet, was a better representation of the needs for the ABDAR Demonstration System. Intranets and Extranets are often centralized rather than distributed. One way to view the ABDAR Demonstration System is as a Website, containing the containers and components necessary to perform assessments that are accessed through a client applet. (This is slightly contrary to a normal view of the Web, which typically uses a Web browser to access a site. In early iterations of the ABDAR Demonstration System, a browser was used, but it became unwieldy as the functionality on the client side increased. A conversion back to a browser side client should be simple as browser and Java technology become more advanced.)

The ABDAR Demonstration System provided assessment tools through software generated objects coupled with views of the ABDAR assessment database. These tools were stored in a central location (an ABDAR Server) and were delivered to the assessor through a client (an ABDAR PMA) over a controlled Intranet. The assessor requested, used, and manipulated these objects and tools in performing the assessment process. This approach promoted modularity and reusability within the ABDAR Demonstration System, with the potential for use throughout the entire maintenance environment. When implementing an ABDAR Demonstration System type unit into a weapon systems' IMIS, the developers of the IMIS can choose which objects best enhance their system. IMIS developers can also choose which objects are redundant or already provided by their system and implement the object accordingly. The IMIS need only support the necessary view of data for the object to work properly. This view can potentially be generated through middleware, database mediators (as in the case used in the ABDAR Demonstration System with Enterprise Java Beans [EJBs] and Tengah Application Server), or direct access to an IMIS database. Figure 8 – Component View illustrates how the assessment tools were integrated within the system.

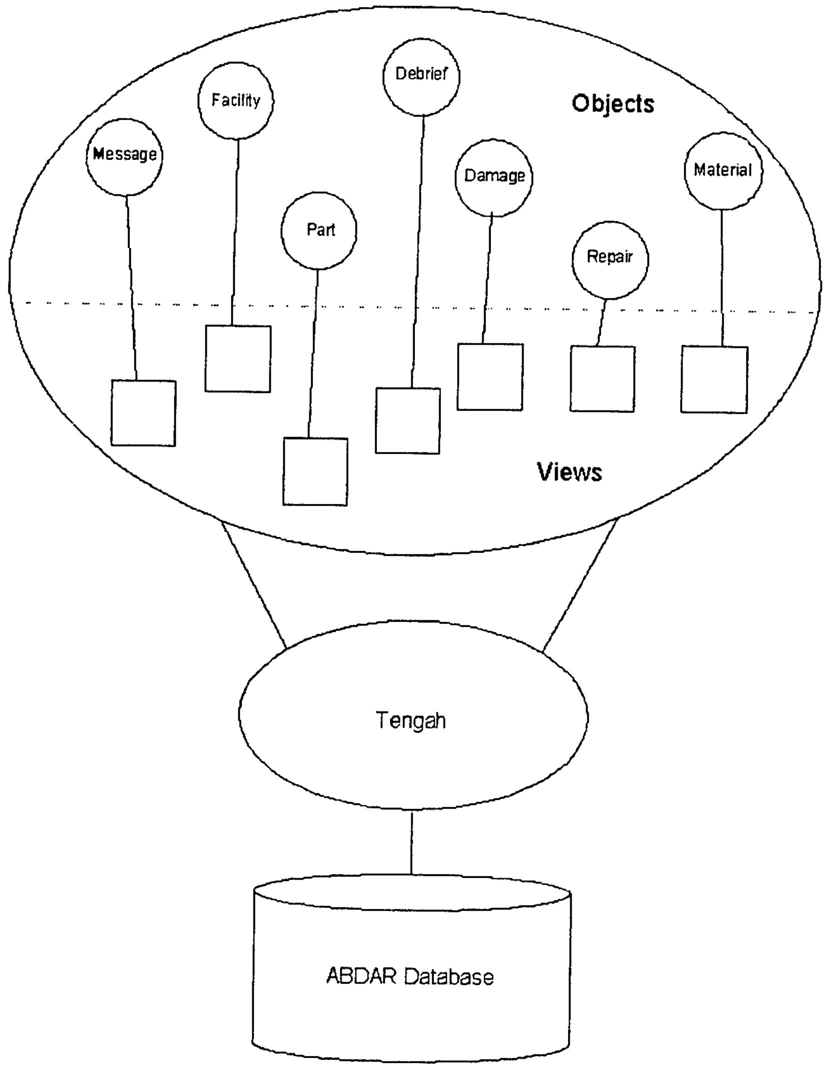


Figure 8 – Component View

The basic components used in the assessment portion of the ABDAR Demonstration System were divided into three major groups, Assessment Objects, Resource Objects, and Miscellaneous Objects.

a. Assessment Objects.

1. Assessment objects contained the following components:
   - (a) Organization
   - (b) Aircraft
   - (c) Flight Schedule
   - (d) Discrepancy
   - (e) Discrepancy Trigger
   - (f) Fault Report

(g) Action
(h) Activity
(i) Debrief
(j) Damage Site
(k) Damage
(l) Unexploded Ordnance (UXO) Inspection
(m) Repair
(n) Viewer

2. Each Aircraft had a Flight Schedule that referred to its next scheduled mission. That mission was either Air to Air or Ferry. An Aircraft may have one to many Discrepancies, although there was only one open ABDR Discrepancy at a time. Discrepancies required the performance of various Action Sets to be resolved.

3. An Action (known in the database as an Action Set) is something scheduled to be performed during the ABDAR process. An example of an Action Set would be the Walkaround. The Walkaround was assigned to an individual, had a planned start time and stop time, did consume or utilize resources, and had a list of tasks or activities that needed to be performed for the Walkaround to be completed. There were several types of Actions, including Debrief and Walkaround Actions (one of each for every Discrepancy), Zone Actions (there were 14 discrete identified Zones on the F-15), Damage Site Actions (there may be zero to many Damage Sites identified on each Zone), and Repair Actions (all possible Repairs were identified, but only certain ones were selected).

4. An essential piece of each Action is the Activity (or in the database, Activity Item). The Activity represented the tasks that were performed during each Action. For example, the Walkaround Action contained two Activities: a UXO Inspection (an Activity of type UXO Inspection) and a high level evaluation of the aircraft (an Activity of type Walkaround). Each Activity contained instructions (a CDM task or IPDF link) and collected information (size of a hole, results of an inspection, authorized signature, etc.). Activities had one other important quality or behavior, they identified other Actions or Activities that needed to be performed. For example, the purpose of performing the Walkaround Activity is to identify Zone Actions (or evaluations) that need to be accomplished. Again, there are various types of Activity Items. A Debrief Activity corresponds to a Debrief Action, Walkaround Activities to Walkaround Actions, Zone Activities to Zone Actions, Damage Site Activities to Damage Site Actions, and Repair Activities to Repair Actions. The sole exceptions here are that UXO Inspection Activities and Damage Activities have no associated Actions.

5. There is a very hierarchical nature of Actions and Activities. They are interdependent, and so the data that builds them was challenging to construct.

6. When an Aircraft was identified with an ABDR Discrepancy, automatically a Debrief Action and a Walkaround Action were created. The Debrief Action automatically created a child Debrief Activity, and the Walkaround Action automatically created both a child Walkaround Activity and a UXO Inspection Activity. The UXO Inspection Activity had to be performed before the Walkaround Activity could be marked "complete".

7. During the Walkaround Activity, Zones were identified that contained damage and Zone Actions were assigned to an Assessor to accomplish. There could be only one Zone Action for each damaged Zone, no matter how many individual damages were identified in the Zone. Each Zone Action automatically created a Zone Activity. At the end of the Walkaround, the Action complete time was recorded, and the Activity was recorded as completed by the Walkaround Assessor.

8. During the Zone evaluation, one or more Damage Sites were identified within the Zone, and Damage Site Actions were assigned to an Assessor to accomplish. Each Damage Site was sequentially numbered, and a separate Damage Site Activity as well as a UXO Inspection Activity was created for each. At the end of the Zone evaluation, the Zone Action complete time was recorded, and the Zone Activity was recorded as completed by the Zone Assessor.

9. Before the Damage Site Action could be accomplished, the UXO Inspection Activity had to be performed successfully. When an Assessor began performing a Damage Site assessment, one or more Damage Activities were created. Upon completion of the Damage Site Action, the completion time was recorded and the Damage Site Activity was marked as completed by the current Assessor.

10. Damage Activities spawn one to many potential Repair Actions, depending upon the technical data. If CDM data was being used, the Repair Actions and Activities were generated automatically. If IPDF technical data was being used, the assessor had to manually record all relevant Repair Actions.

b. Resource Objects.

1. Resource objects contained the following components:

(a) Resource Requirement
(b) Resource Allocation
(c) Resources
(d) Resource Order

(e) Equipment
(f) Facility
(g) Material
(h) Part
(i) Personnel

2. A Resource is any type of consumable or non-consumable used in or by the ABDR process. Equipment, Facilities, Material, Parts, and Personnel are all types of Resources. Each type of Resource has a name, an availability status, and a unique identifier. Non-consumables could be checked in and out by authorized personnel, and consumables had on-hand volumes that could be incremented and decremented, as appropriate. Resources could be identified as being required for the performance of a Repair Action, but identifying a Resource did not commit that Resource. Next, when a Repair was selected, a specific Resource could be reserved (or allocated) by authorized personnel for a specific duration. Specific Personnel were also among the Resources that could be allocated.

c. Miscellaneous Objects.

1. Finally there are a few components, not previously covered, that "round out" the ABDAR Demonstration system:

(a) Codeset
(b) Message
(c) Message Recipient
(d) User Settings

2. Codesets are lists of legal values that may be used for certain fields in the database.

3. The Message and Message Recipient tables support the messaging system in the ABDAR Demonstration System. The reason for two tables is that conceivably, a single message might be sent to more than one recipient, but there is no reason to duplicate the message in multiple records. Dividing things into two tables allowed the message to be written once, but to be sent to multiple users. Additionally, the "read flag" in the Message Recipient table allowed the system to determine whether or not the message had ever been read by the recipient. If it had not, it was displayed on the Home Page as a new message.

4. User Settings was designed to allow the ABDAR Demonstration System to "remember" what aircraft the user was working. As soon as an Aircraft had been selected, it was written to this table along with the user name. The next time the application was opened, the Aircraft was pre-selected. The setting was also referred to at times during the ABDAR session to pre-select some information.

Software Development Environment

The Software Development Environment chosen supported the needs of a typical Internet project. Since the Internet evolved as the project progressed, the Software Development Environment had to evolve with it. The environment listed below contains the final set of tools that were used to build ISD #4.

a. Server Environment

   1. Database Server – Oracle Enterprise Server 7.3
   2. Application Server – WebLogic Tengah 3.1.3

b. Web Development Environment

   1. Java Developer's Kit (JDK) 1.1.7
   2. Visual Café for Java Professional Edition 3.0
   3. HTML Development – Word 97 for Windows
   4. Enterprise Java Bean Development – WebLogic Tengah 3.1.3

User Interface

The user interface was comprised of software containers or pages. These containers mapped to the major ABDR functions or phases identified during the IDEF3 modeling process. The overriding philosophy was that during each phase of the ABDR process, the assessor would be able to go to a page and find the tools necessary to perform the appropriate function. Primary functions supported in the ABDAR Demonstration System were Debrief, Initial Inspection, Detailed Inspection, Repair Selection, and Repair Planning. Homepage, Library, and Documentation pages were added to support an electronic version of ABDR. Additionally, the user was provided with access to the WI program and the technical data viewer, CDM or PDF.

An essential component of the interface was the icon set used for navigation. Many of the icons were developed using a revolutionary technique for developing meaningful symbols. Potential users of the system were brought together for focus groups. The focus groups were presented with symbol scenarios representing the icons to be developed, such as, Debrief, Repair Planning, Equipment, Tools, etc. Symbols developed in these groups were then compared to symbols developed by individuals working alone. The symbols developed by the focus groups tended to be more meaningful than the symbols developed by the individuals, as ranked by potential users in an evaluation task.

Before the user interface was finalized, a late prototype was submitted for Heuristic Usability Testing. Heuristic Usability Testing means that a select group of users evaluates the interface for usability issues, large or small. In this case, two users closely involved in the design process were brought in and solicited for comments. The feedback received, from the two users, contributed significantly to the final user interface used in ISD #4.

The use of focus groups to develop the icons, as well as the Heuristic Usability Test group, provided integral information needed to develop a usable interface. The icons developed were meaningful and the interface received praise from the final AUG members and subjects in the field test. In summary, both methods were successful in accomplishing the shared goal of a user-friendly interface.

During Heuristic Usability Testing, it was discovered that training would be an issue for first-time users of the ABDAR Demonstration System. Familiarization of the assessment tools could not be accomplished without some form of formal training. To alleviate the amount of practical training required, data specific tutorials along with software Wizards were developed to assist the user.

Difficulty with the Java Swing package, middle-tier and database artifacts, and a lack of consistency between visual objects precluded the development of the perfect user interface.

Home Page

The Home Page was designed to orient users to the ABDAR Demonstration System. After a successful logon, the Home Page displayed a list of the user's current assignments and automatically populated the Messages Area with unread messages (see Figure 9).

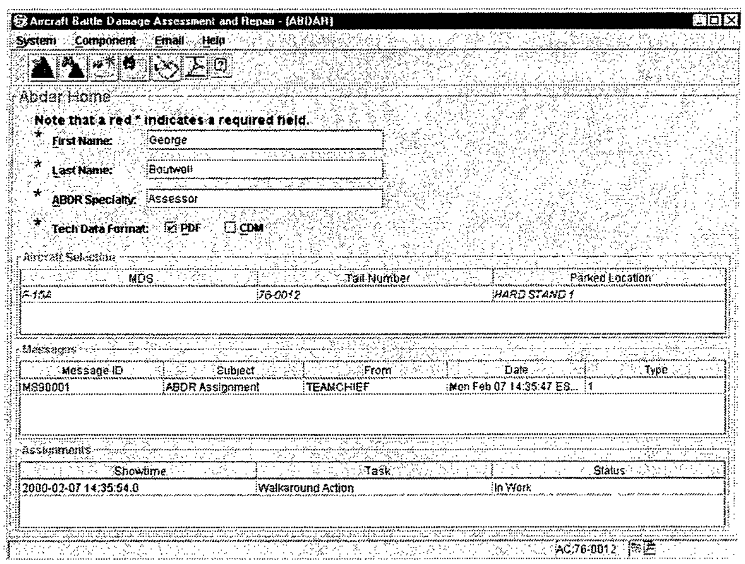


Figure 9 – Home Page

The Assignments Area on the Home Page was populated (from the database) with all the tasks assigned to the user. To ensure continuity, the aircraft tail number was displayed on the right side of the tool bar. If an aircraft had not been previously selected or if a different aircraft was desired, the user could select an aircraft from the Aircraft Selection Area. This was accomplished by clicking on it, which then populates the Assignment Area with the assignments for that aircraft.

Debrief

The Debrief page collected information as required by TO 1-1H-39, from the aircrew and presented it on the ABDAR Demonstration System for use by the users (see Figure 10). Additional information may include Fault Codes occurring at time of incident.

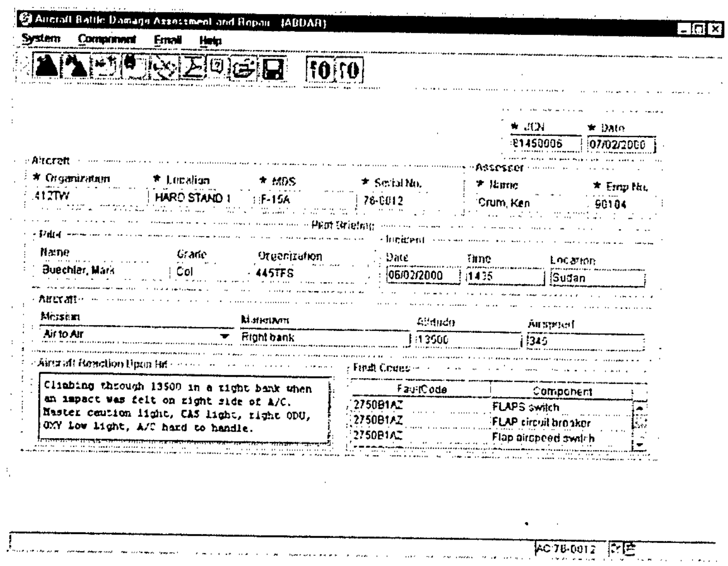


Figure 10 – Debrief

If information had been downloaded from an external source, such as CFRS, the information collected appeared automatically when the Debrief page was opened. A Create function allowed for population of the Debrief form without the help of an external debriefing module. This included populating the AFTO Form 781A data upon selection of the submit function. The Submit Function sent information to ABDAR database.

Initial Inspection

Initial Inspection is analogous to the assessor's initial Walkaround. The user must perform an initial UXO inspection and Safe the aircraft. Then, zones with damage are identified, and damaged areas within those zones are further identified. Each time a damaged area within a zone was identified, that damage was created with a sequential number, Damage 1…n and the user had to inspect that damage site for UXOs.

a. In the CDM version, most of the user interaction was with the imbedded CDM viewer. After the user entered prompted information, it was reflected in state changes appearing in the forms (see Figure 11).

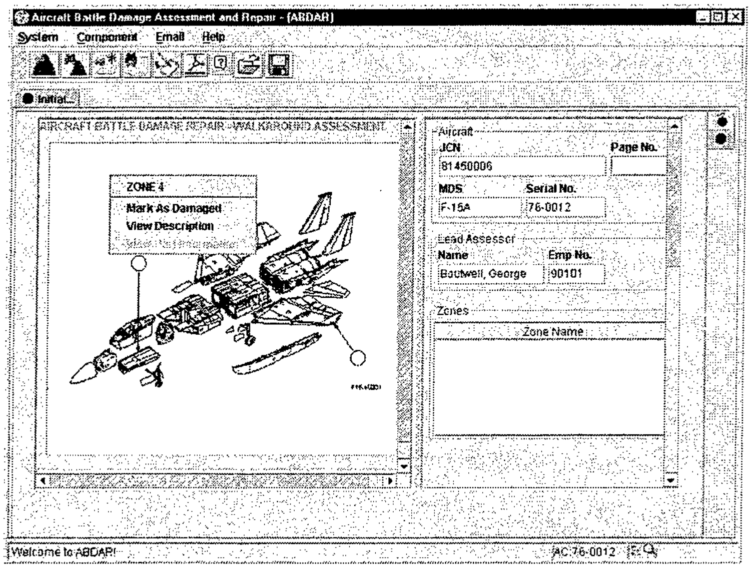


Figure 11 – CDM Initial Inspection

b. In the IPDF version, the viewer (Adobe Acrobat) and the forms acted independently of each other (see Figure 12). This led to the implementation of an IPDF Damage Collection Wizard process. This wizard process made the IPDF Damage Collection activity easier to use and navigate.

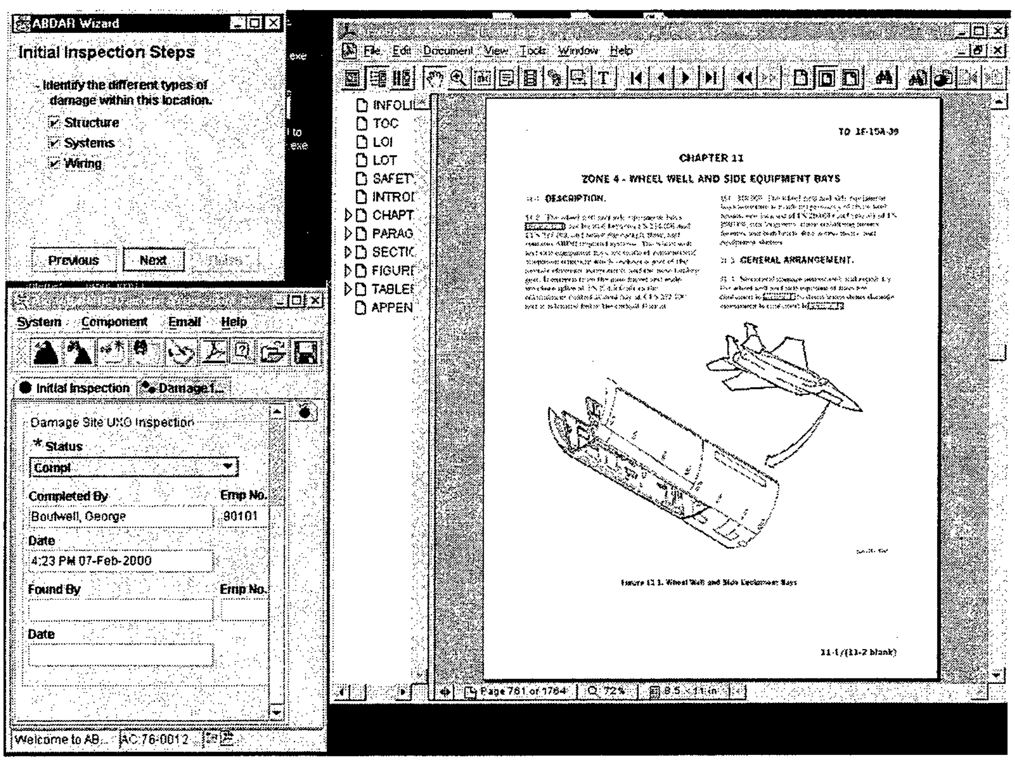


Figure 12 – IPDF Initial Inspection

Detailed Inspection

The damaged sites created during the Initial Inspection and numbered are evaluated further (see Figure 13). For example, in the CDM Damage Collection version, a damage site is evaluated for system or structural damage. In the IPDF Damage Collection version, the user is asked what types of damage can be identified in that damaged area such as system, structure, and wiring. Damaged components underneath each type of evaluation are created and repairs identified (either manually via IPDF or automatically with the CDM data).

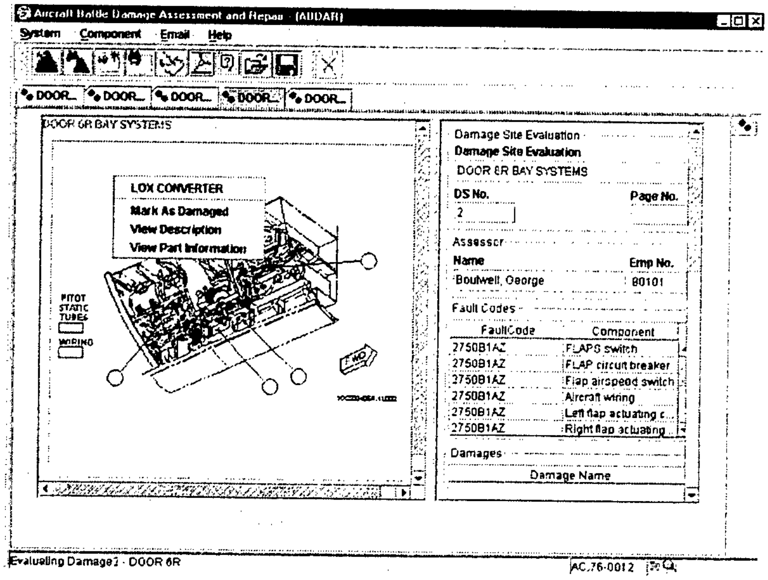


Figure 13 – CDM Detailed Inspection

Repair Selection

Repair Selection provided information allowing the user to make intelligent choices about which repair to perform (see Figure 14). This container brings all those variables into focus, allowing the best decision to be made. The user was able to view all the potential repairs identified in Detailed Inspection. Additionally, the user could populate the resources needed for those repairs, as well as other information such as Estimated Time To Repair (ETTR), Flight Restrictions, Mission Essential Subsystem List (MESL) impacts, and a preview of the repair.

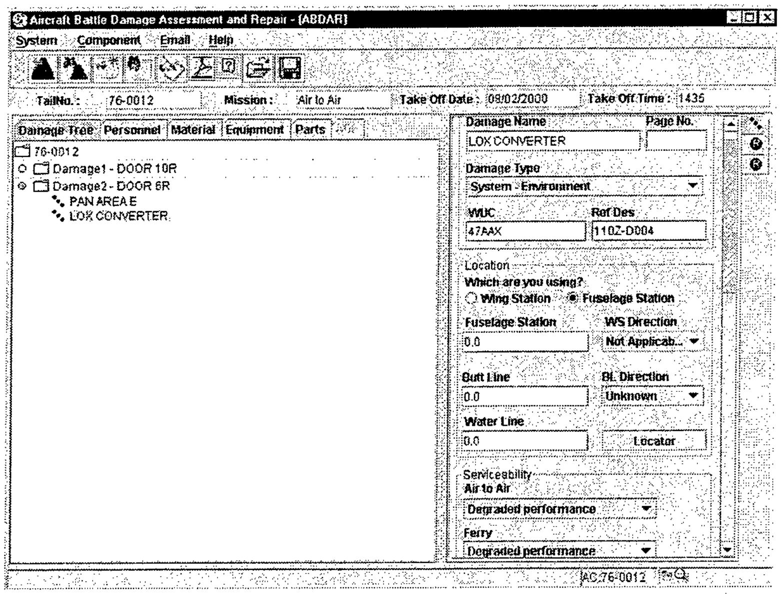


Figure 14 – Repair Selection

Repair Planning

Repair Planning contained a timeline view of the aircraft evaluation and repairs (see Figure 15). This timeline was capable of switching to a simulated view of the different resources assigned to those tests and repairs, so the assessor or supervisor could spot potential bottlenecks and properly schedule resources. The user could select any evaluation or repair and assign resources (people, equipment, parts, material, etc.) to the action.

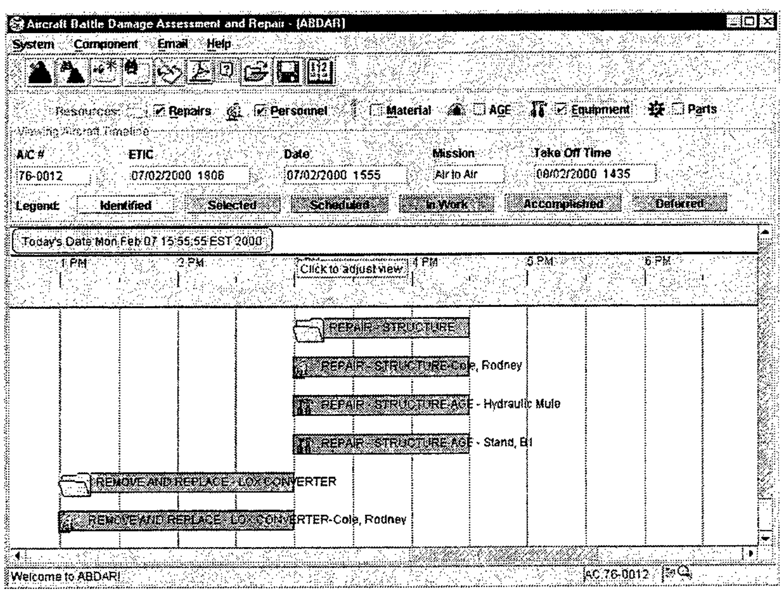


Figure 15 – Repair Planning

Library

Library provided a tree-like (Table of Contents) view on the left-hand side, and a panel for data viewing on the right-hand side (see Figure 16).

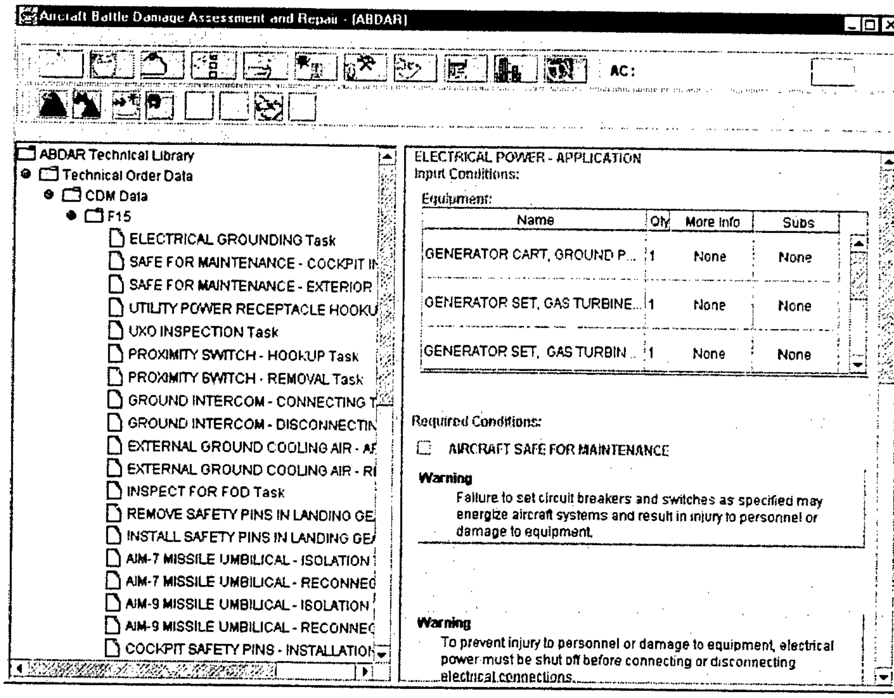


Figure 16 – Library

a. If the user was viewing CDM data, the left-hand (Table of Contents) contained a system and subsystem hierarchy of the CDM data set. The right-hand side provided a view of the tasks, descriptive information, or part information the user selected.

b. If the user was viewing IPDF data, the left-hand side contained a list of all IPDF TOs available. Clicking on a TO name enabled the Adobe Acrobat viewer to switch to the corresponding IPDF document.

Documentation

Documentation provided an electronic view of the paper documentation for an aircraft (see Figure 17). Any maintenance accomplished must be documented on the AFTO Form 97.

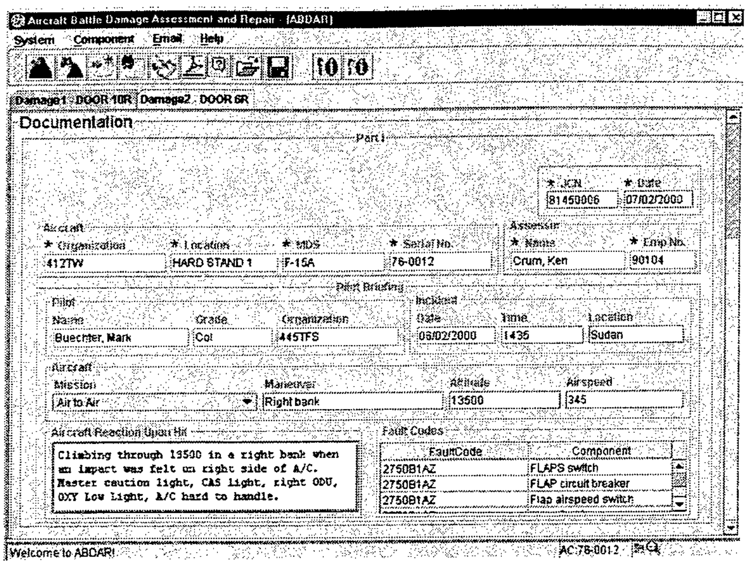


Figure 17 – Documentation

Wizards

The ABDAR Wizards were a set of instructions provided to the user as an online assistant for using the ABDAR Demonstration System (see Figure 18). The Wizard was automatically turned on when the ABDAR Demonstration System started, but its use was optional in most screens (the exception being the IPDF Initial and Detailed Inspection). The Wizard was always available by clicking the Wizard Icon on the ABDAR Toolbar. The purpose of the Wizard was to guide the user through the ABDAR Demonstration System assessment process, one step at a time. By using the Wizard, the assessor ensured that the assessment was complete and accurate.

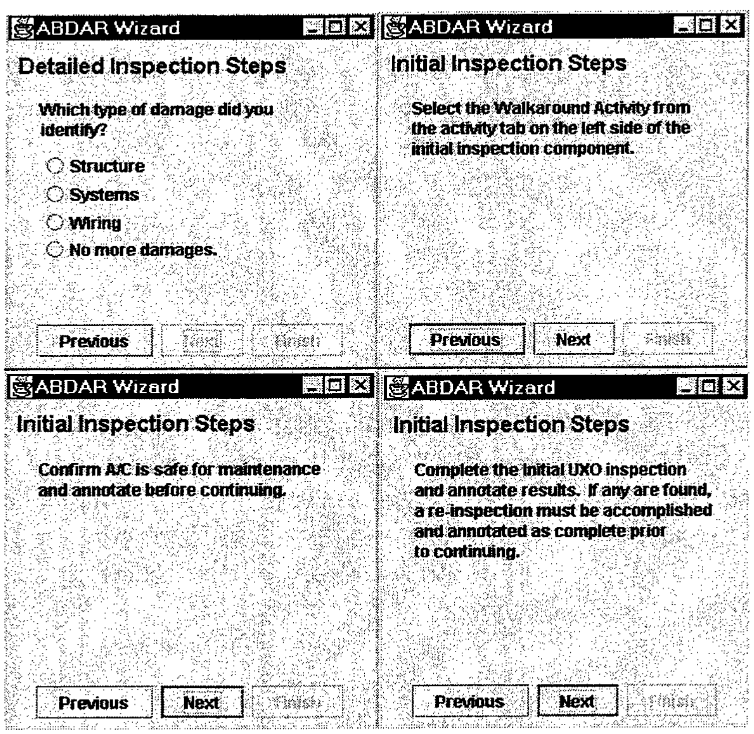


Figure 18 – ABDAR Wizard Examples

The ABDAR Wizard was composed of a series of "hard coded" Java panels. To navigate through the wizard panels Next, Previous, and Finish buttons were provided. The Next button was used to view the next step in the process; the Previous button was used to review a step; and the Finish button was used when the end of an ABDAR component had been reached. If a function should not be performed at some stage of the Wizard process, the buttons were grayed out. For example, if critical data had been entered into a Wizard Panel and saved to the database, a user could not back-up to that panel and change the data.

A Java interface was provided to the ABDAR Wizard that allowed it to communicate with the ABDAR Demonstration System. The interface provided methods which allowed the user to input information directly through the ABDAR Wizard. This design was beneficial for the IPDF version and reduced the amount of training necessary. However, this design was not as beneficial for the CDM version, since most of the information was already contained in the CDM data.

Wiring Illuminator (WI) / (DWDS)

The WI is a commercial software application purchased for use in the ABDAR Demonstration System that allowed for quick and easy access to wiring information. It was used in detailed inspections to assess wiring damages and in Repair Accomplishment to repair the wiring.

The design of WI was modular and configurable. Adding windows with different views of the wire data was easy with the WI design. In addition, the data source could be flat files or a database system. For the ABDAR Demonstration System, an instance of the Oracle database was used.

The WI application, available from Boeing, displays six views of the wire data and provides sufficient information to assess and repair damaged wiring. The name/type of view and a brief description of each view are given below.

a. The Context View. This view displays information in a tree structure used to select the data shown in the other views. It contains a status panel that provides for entering and displaying current context information. The Context View also contains a menu bar to exit WI, save the current window layout, open/close the other views, or obtain help.

b. The Wire Diagram View. Displays a wiring diagram of the reference designators (RefDes), pins, and wires in the current circuit or wire run. It also contains a status panel that displays wire information.

c. The Wire List View. Displays the same information as the Wire Diagram View, but as a tabular list.

d. The Pin List View. Displays a tabular list of all of the pins and connecting wires for the current RefDes.

e. The Locator View. Displays a graphic of the location of the current RefDes on the aircraft.

f. The Pin Arrangement View. Displays a graphic of the pin arrangement of the current RefDes.

A "Bundle Assessment" enhancement was added to the WI for the ABDAR Demonstration System. This enhancement consisted of a Bundle Assessment Graphic view and a Bundle Assessment View, which are described below.

a. Bundle Assessment Graphic. These graphics displayed the wiring bundles as selectable segments. The assessor used these graphics to identify the segment of the bundle that contained the damaged wires. The selection of a damaged segment filtered the wire data for the bundle, to the data contained in the damaged segment.

b. The Bundle Assessment View. This view displayed a list of systems and wires involved in the repair of a bundle segment. This view (see Figure 19) was used during the assessment and repair of the wires and systems of a bundle. The user kept track of the necessary actions to be taken, with respect to these wires and systems, by selecting and setting the "status" field in bundle assessment view. The Bundle Assessment view was divided into three sections: The System list, the Wire list, and current bundle information. The System list section showed systems affected by the selected bundle segment, which were to be assessed and repaired. The Wire list section showed only wires in the system, which was part of the affected bundle segment. The current bundle information section showed the current bundle. All the information in the three sections was obtained either from a previous assessment or from a selection from a Bundle Assessment graphic, which was provided in the default ABDAR view of WI.

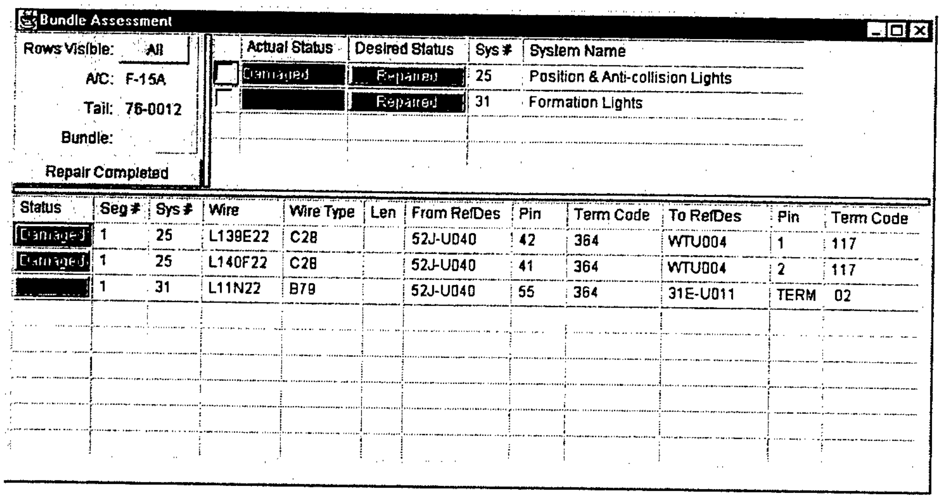


Figure 19 – Bundle Assessment View

c. Selecting the left column of the System list, and setting the "Rows Visible" property to "Marked" displayed, in the wire list, only the wires for the systems that were selected. Setting the "Rows Visible" property to "All" showed wires for all the systems in the damaged segment of the bundle.

d. Pressing the buttons in the column labeled "Desired Status", on the System list, set the different actions possible with respect to the particular System.

e. Pressing the buttons in the column labeled "Status", on the Wire list, set the different actions possible with respect to the wires. At the same time, the "Actual Status" for the System list changed automatically.

f. Clicking on a RefDes in the Wire list triggered a context change. All current views changed to reflect that RefDes. For example, the Locator View updated to show the current location of the selected RefDes.

g. Pressing the button labeled "Assessment Completed" saved the information on the lists to the database or flat file. Once the assessment was completed, the button name changed to "Repair Completed". This button also saved the information on the lists to the database or flat file.

Technical Order Presentation System

In the context of the ABDAR Demonstration System, a presentation system is a software component (Viewer) used to present technical order data. The ABDAR Demonstration System used a CGM viewer (built in-house) to present CDM data and a PDF viewer (Adobe Acrobat Exchange) to present IPDF data.

CDM

When the ABDAR Demonstration System was developed, there was no third party CDM Viewers available that could adequately handle the system requirements. NCI produced an in-house version that executed with the ABDAR Demonstration System. In developing the CDM viewer, methods were chosen for storage, retrieval, and presentation of CDM data. In choosing this design, the decision was made to develop a Viewer that performed the functions necessary for the ABDAR Demonstration System. The CDM Viewer used in the ABDAR Demonstration System was not designed to work with other information maintenance systems.

In implementing a schema for storing the CDM data, NCI relied heavily on a design used by Boeing for the Apache presentation system. The data was exported from the Quill authoring system into an SGML file stored in ASCII format. Within the contents of the SGML file existed the pieces of data necessary, ideally, for an entire weapons system's set of assessment data (for the field test, a subset was used). Each <system> element in CDM data could contain five other elements of data, with infinite cardinality. The five types of elements were the tasks, descriptive information, fault isolation information, part information, and other system elements. Any element, which was subordinate to a system element, was an element that was pertinent to that system. Since system elements can contain system elements, the CDM storage structure had to be recursive in nature. Additionally, each of the elements mentioned above could have their own set of subordinate elements. This schema would generally be represented by a graph data structure with 1 to *n* nodes. For the ABDAR Demonstration System, a tree structure was used which stored a system element that represented the aircraft.

The SGML file was imported into an Oracle database for use by the ABDAR Demonstration System because extracting data from a relational database was quicker than parsing the data real-time from a single flat-file. Each element had an entry in a table named for its element type (for instance, there was a "task" table, a "step_seq" table...). Additionally, because of the hierarchical nature of the data, a single table called "sub-components" was created which uniquely identified every CDM element, its parent elements, and a list of child elements, where appropriate.

Rapid retrieval of the information from the database to the viewer component was instrumental in the success of the ABDAR Demonstration System. This required the ability to efficiently traverse the sub-component table. A "depth-first traversal" algorithm used for adjacency matrixes was used for traversal. Since the sub-component table was not necessarily in a format that could utilize this type of algorithm, some pre-processing of the data was done upon system start up. A "preloadCDM" function was developed that selected components from the Oracle database and stored them dynamically in an Enterprise Java Bean. As the CDM Viewer needed data to display to the user, it invoked remote methods on the EJB to request the proper CDM elements. This design provided a secondary benefit; the EJB was executed on the server platform thereby distributing the processing between two processors (unfortunately, this benefit was mitigated because of the relatively small connection rate imposed by the RF connection).

The CDM viewer displayed three types of tasks to the user: procedural tasks, assessment test tasks, and other test tasks. The CDM Viewer would automatically change its mode based upon the type of task it was displaying and the context for the viewer. A Java interface was provided to allow communication between the CDM Viewer and the ABDAR Demonstration System. For example, a setClass() method was provided through an API to allow the CDM Viewer to set the class of a structural damage. The interface gave the appearance of a highly coupled system in which the user could interact almost exclusively with the CDM Viewer during Damage Collection.

IPDF

For the IPDF version of the ABDAR Demonstration System, the electronic TOs were stored in a flat file IPDF format. PDF is a file format, created by Adobe, that lets you view and print a file exactly as the author designed it, without needing to have the same application or fonts used to create the file. Since its introduction in 1993, PDF has become an Internet standard for electronic distribution that faithfully preserves the look and feel of the original document complete with fonts, colors, images, and layout. IPDF data is PDF data that has internal *linking* or *indexing*.

Adobe Acrobat Exchange was used to view IPDF TO files. It was comprised of Acrobat Reader and two Acrobat plug-ins, Acrobat Search and AutoIndx. Acrobat Reader is the tool used to navigate and view the IPDF files. Acrobat Search was used to search a collection of files stored in an index file created by Acrobat Catalog. Exchange ran as a platform specific (Windows) executable program separate from the ABDAR Demonstration System. A custom plug-in was developed with the Adobe Acrobat Software Development Kit (SDK). This plug-in enabled a Dynamic Data Exchange (DDE), so the ABDAR Demonstration System could manipulate and retrieve TO name and page number from Exchange. As the user identified damages and repairs, the ABDAR Demonstration System stored the file name and page number that the user was viewing. The ABDAR Demonstration System could then force Exchange to load the saved page any time the user selected the corresponding damage or repair. Additionally, a few pages were bookmarked and loaded automatically, such as the zone breakdown page and the locator graphics. Any additional information needed for the assessment process had to be copied from the IPDF file by the user and stored in the appropriate ABDAR Demonstration System supplied form.

Software Architecture Overview

An *n*-tier architecture was used for development of the ABDAR Demonstration System. This architecture supported the Web-based approach and component architecture previously highlighted. The Application Server Architecture provided services for connecting the Client to the database. The Client Application Architecture supported and delivered the GUI pages.

Application Server Architecture

The application server was populated by problem-specific logic, commonly called business objects. In the Enterprise Java Bean framework, these business objects are broken into two basic types: Session Beans and Entity Beans. Session Beans provided services but did not have persistence, while Entity Beans had persistence. Both types executed within an application, known as the Enterprise Java Bean Server. This server provided the business objects with transaction services, distributed events, and state management.

The ABDAR Demonstration System used an Enterprise Java Bean Server from BEA/Weblogic, known as the Tengah Server. Within the Tengah Server, the business objects were broken into two layers: an Entity Bean and Session Bean layer. The Entity Bean layer provided business rules and persistence. The Session Bean layer provided service to the client applications, maintained a user's session, and delivered data packets to clients. Figure 20 – Tengah Application Server, shows the flow of data from the database to the client application. The design achieved its flexibility by decoupling the upper layers from the lower layers. Thus, the business rules could change within the Entity or Database layer with little affect on the Session or Client layers.

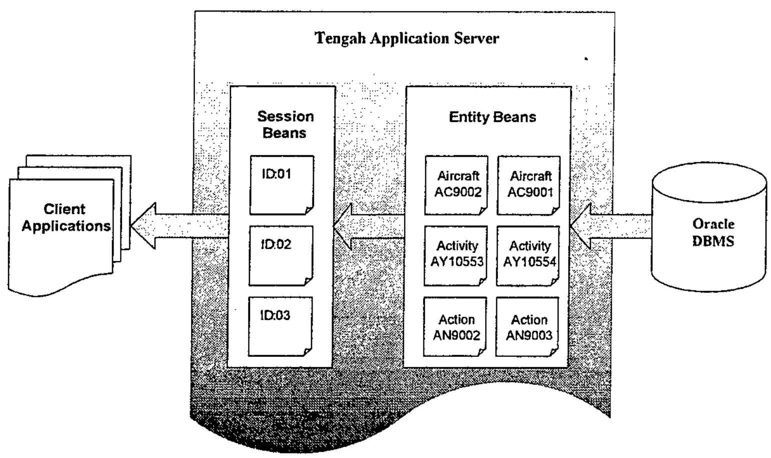


Figure 20 – Tengah Application Server

Client Application Architecture

The goal of the client application architecture was very similar to that of the application server architecture, to achieve flexibility by decoupling the upper layers from the lower layers. The client application layers included the main controlling applet/application, the ABDAR Demonstration System containers/forms, and the client models. The main controlling applet/application was responsible for displaying the desired container and providing global tools such as alerts, notes, and mail. The ABDAR Demonstration System containers/forms contained all the presentation logic associated with the client application and allowed the user to input information into the

system. The client models were responsible for establishing connection to the remote session beans and maintaining the system state.

Figure 21 – ABDAR Client Application, shows the flow of user input through the system. The presentation logic, which was in the Client Application, could change with little affect on the server application or the client models.

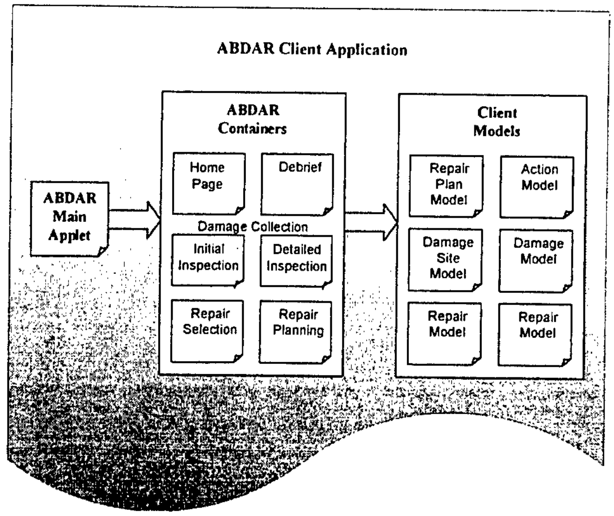


Figure 21 – ABDAR Client Application

Database Architecture

A relational database was used to support the ABDAR Demonstration System. Its responsibility was to maintain persistence of all the assessment server objects and to store the CDM and WI data. All connections to the database from the client used the application server as a conduit. Oracle Server 7.3 was the application that ran all the software associated with the database. For each installed version of the ABDAR Demonstration System, Oracle was installed as a single server, but there were many database instances running against a single installation of Oracle. A database instance is defined as set of server processes, a group of users and a collection of tables, triggers, synonyms and other database objects.

Instances

Objects within the database were owned by the user account that created them. When modifying any of these objects, it was important to be logged in as the appropriate user. Within each instance, there were three primary ABDAR Demonstration System accounts that owned database objects:

a. DWDS: Owned all tables, constraints and data created by Boeing to support the DWDS subsystem.

b. IETM: Owned all CDM tables, constraints and data along with local tables created to support the CDM viewer portion of the ABDAR Demonstration System.

c. ASSESS: Owned all tables, constraints, triggers, procedures and data created to support the damage collection and assessment portion of the ABDAR Demonstration System.

The ABDAR Demonstration System database had both a physical and a logical architecture. Physically, the database was laid out as a multi-process, multi-threaded application. Logically, the data was stored in data files spread across three logical partitions in the connected mode and one logical partition in the stand-alone mode.

Logical Organization

The DWDS and IETM instances were considered portions of the WI and Viewer components. The instance of most interest to the ABDAR Demonstration System was the ASSESS instance. Its logical organization was similar to the component modules. Data was contained in Tablespaces (and a tablespace may be mapped to one or many datafiles). Different tablespaces held different types of data. The ABDAR Demonstration System was set up with the following tablespaces:

a. Assessment Objects

b. Resource Objects

c. Field Test Objects

d. Miscellaneous Objects

For the most part, the ABDAR Demonstration System database objects were passive in that they did not initiate or perform actions. The exceptions were Triggers and Stored Procedures (and Functions). A procedural coding language called PL/SQL (Procedural Language extension to Structured Query Language) was used to develop a small set of procedures that were used when data needed to be generated in the database. For example, a timestamp and unique identifier (ID) were generated whenever a new Action was created. Triggers are pieces of code that are executed automatically by the database when a certain condition is detected. The ABDAR Demonstration System used a few triggers that executed when specific kinds of data were created. Additionally, stored procedures were used that executed on demand rather than automatically. The ABDAR Demonstration System used stored procedures, mostly to reset the data.

Loading and Resetting Data

Because of how the ABDAR Demonstration System database was used, it was very important that resetting the data be very quick and easy. The same set of procedures accomplished the initial load of the data as well as resetting it after it had been used. Three versions of the data could be reset at any time by any authorized user using one of the following procedures described below:

a. LOADDEV: This procedure was used most frequently. It reset the largest set of data for use by the developers and system testers. It called to a set of nested, modular procedures that each reloaded an individual table or pair of tables.

b. LOADLEFT and LOADRIGHT: These two procedures were for use during the field test. Specific sets of data were needed for the evaluations on each side of the aircraft. Like LOADDEV above, they also called nested, modular procedures.

This section, Software Design, detailed the development of a three-tier architecture used to develop the ABDAR Demonstration System (ISD #4). It highlighted the basic constraints NCI confronted. Detailed views of the user interface, middle-tier, and database were presented. The next section, Hardware Design focuses on the hardware platforms necessary to support the ABDAR Demonstration System in the stand-alone and connected modes of operation.

HARDWARE DESIGN

The ABDAR Demonstration System hardware was designed to work in two modes of operation, stand-alone and connected. The stand-alone mode was implemented to support ABDR personnel deploying to remote sites without the capability to connect to a network system. The connected mode was implemented to support ABDR personnel with the capability and availability of a network system. To minimize the expense of the project, NCI put forth a concerted effort to use only off-the-shelf hardware. Therefore, Intel-based laptops and servers running a Microsoft Windows environment were used, although the ABDAR Demonstration System software was platform independent.

Stand-Alone

In the stand-alone mode, the information provided by the system was limited to that information maintained in the PMA. There was no transfer of information with the ABDAR Server or a base network. Information could only be input into the system by a single user or manually loaded into the PMA before the start of the ABDR maintenance activity. Although all ABDR roles could be supported, any information updates had to be input manually. In stand-alone mode, all software components (Client, Application Server, and Database Server) are executed on a single PMA.

Connected Mode

In connected mode, the PMAs and ABDAR Server transferred information via radio frequency (RF)-link. Information was transferred and shared among the ABDAR Demonstration System components. However, there was no information being transferred to or from the base network, so there was no integrated maintenance functionality. Typically, the client software was executed on the PMA, while the application and database servers were executed on the ABDAR Server.

The field test required the users to have the mobility to inspect the aircraft, while at the same time being able to access or input data into the ABDAR Demonstration System. To accomplish this task, a wireless network was created using Proxim's Wireless Local Area Network (LAN). The wireless LAN consisted of an access point and wireless PCMCIA cards, allowing the users to move anywhere around the aircraft and still have connectivity to the system. The wireless LAN provided the users with a secure, high-speed connection (1.6 Mbs) from the PMAs to the system. In order to keep network traffic to a minimum, the wireless LAN was not connected to any other LAN's or Internet connections.

For the field test the wireless LAN consisted of two servers and five laptops. The two servers, running WindowsNT Server 4.0, were configured as a Primary Domain Controller (PDC) and a Backup Domain Controller (BDC). This architecture was used to ensure 24/7 access to the system in the event that a server failed. The servers were connected to the network using Cat-5 network cables connecting to a Linksys 8-port hub. The five laptops were connected to the LAN using Proxim's RangeLAN2 7400 PCMCIA network cards and a RangeLAN2 Access Point Model 7520. The access point was connected to the network using a Cat-5 network cable connecting to the 8-port hub. The networking protocol used on the wireless LAN was TCP/IP. Each device was assigned a static IP address to allow communication.

Table 2 lists the hardware components used at the field test.

Table 2 – ABDAR Field Test Hardware Requirements

Equipment	#	Configuration Requirements	Applications/Accessories
Dell Latitude Notebooks with WindowsNT Workstation 4.0 (ABDAR_PMA1 – 5)	5 (2 for subject use, 2 for FT admin, and 1 for back-up)	256MB 6 GB HD Win 95/NT with Pentium class processor	Java Runtime Environment 1.1.7a Adobe Acrobat Exchange 3.01 (for PDF viewing) ABDAR Java components
Dell PowerEdge 4200 Server with WindowsNT Server 4.0 (ABDAR TEST) (Ran application and database servers)	1	256MB Five 4 GB HD Win NT with more than one Pentium Pro or Pentium II processors	Application Server (Tengah 3.1) Database Server (Oracle Enterprise Server 7.3) ABDAR Java components
ABDAR Admin Workstation (Test Team Support)	1	256MB 6 GB HD Win 95/NT with Pentium class processor	Java Runtime Environment 1.1.7a Microsoft Office 97 Suite PDF Viewing Capabilities (Adobe Acrobat Exchange 3.01) Database Management Capabilities (Oracle Enterprise Manager) Application Management Capabilities (Tengah Manager) ABDAR Java components
Dell OptiPlex GXMT 5166 Workstation with WindowsNT Server 4.0 (ABDAR_PDC_FT)	1		
Dell Dimension XPS T450 with Windows98 (ABDAR_BDC_FT)	1		
100 MB Zip Drive (External)	1		
RangeLAN2 Access Point Model 7520	1		
Proxim RangeLAN2 7400 Wireless network cards	1		
Linksys 10-port Hub	1		
10 foot network cables	5		
100 Foot cable	1		
25 foot network cable	1		
7 foot network cable	1		
9.0 DBI Elliptical Antenna with bracket	1		
10 foot Antenna Cable	1		
Universal Mounting Bracket for antenna	1		
HP LaserJet Printer	1		

A workstation was used to connect to the Robins AFB LAN, providing field test administrators an Internet connection. Any electronic communication between the ABDAR Demonstration System and the workstation was accomplished using 100mb Zip Disks. The 100mb Zip Disks were also used to transport critical system files from NCI to the field test office at Robins AFB. A HP LaserJet printer was also attached to the workstation to allow the administrators to print important test documents. The hardware configuration is illustrated in Figure 22.

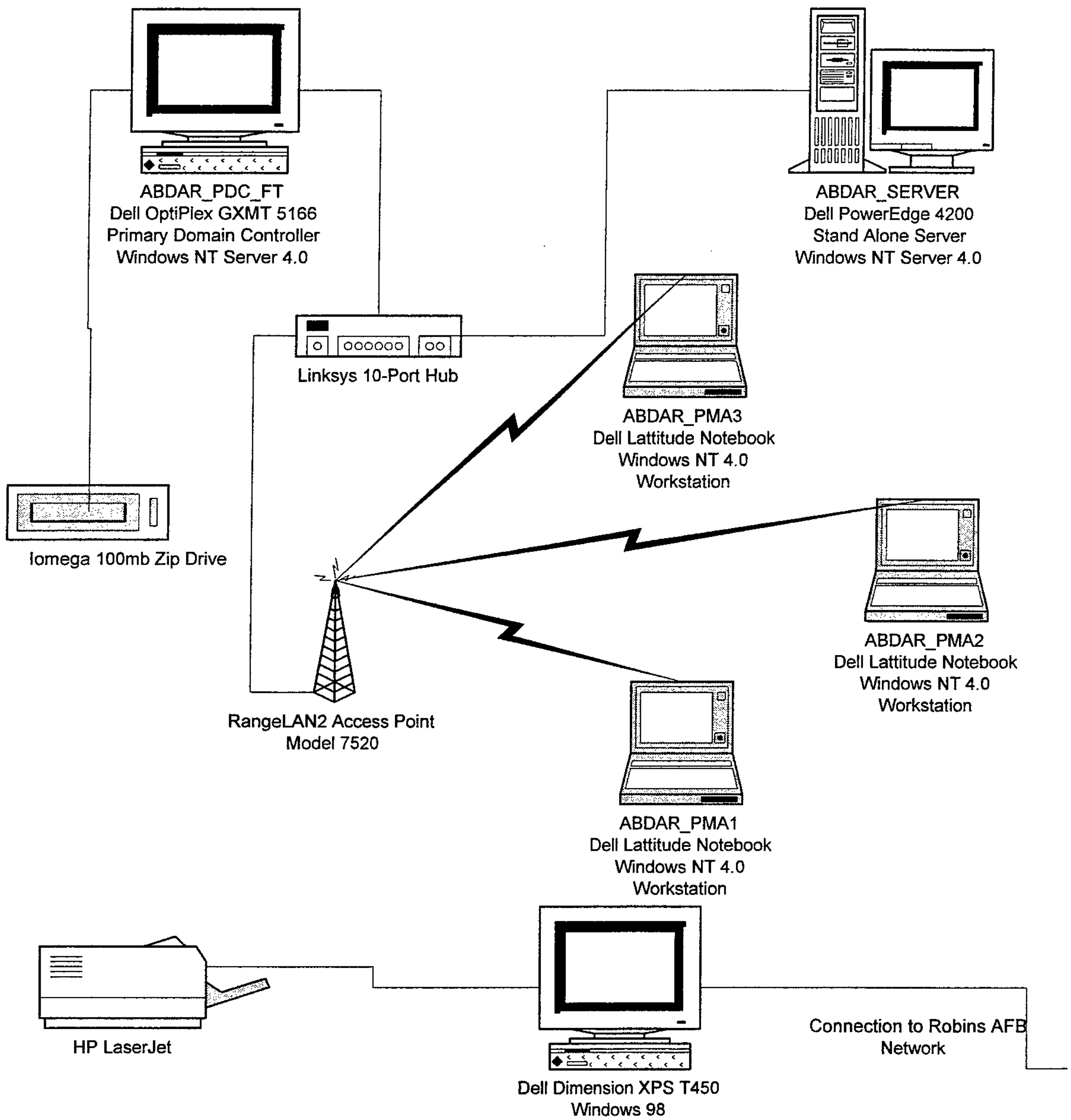


Figure 22 – Field Test Hardware Design

This section, Hardware Design, focused on the hardware platforms that supported the ABDAR Demonstration System in the stand-alone and connected modes of operation. The next section, Data Development, presents the processes necessary to develop data for the ABDAR Demonstration System. The data was subdivided into three types, assessment system, IPDF/PDF, and CDM.

DATA DEVELOPMENT

Sufficient data, as determined by the SMEs, was developed to support the field test. The developed data was divided into three different types, assessment system, IPDF/PDF, and CDM. The assessment system data was the data needed to support functionality (such as inventories of Resources, Debrief information from CFRS, WI wire data tables, aircraft status information, etc.). Assessment system data included all developed data, other than the CDM or IPDF technical order data, that were stored as assessment and resource objects in the Oracle database. The technical order data in ETI format (IPDF/PDF and CDM) supported assessment of damage in the Door 6R and the Left Wing Trailing Edge areas of the aircraft used in the field test.

NCI and Boeing developed the data. Boeing was responsible for the initial development of the ETI along with data needed for the WI operation and data needed to support debrief by CFRS. NCI was responsible for the development of Test Data sets used in software development, system data to support resource management, and aircraft status data. NCI was also responsible for enhancements, integration, and testing of the ETI after Boeing completed the initial development.

In addition to the CDM data required for the field test, the development and coding of the ABDAR Demonstration System depended upon having sample data sets during the development of each ISD. To fulfill this requirement, NCI created test data sets using an authoring environment identical to that used by Boeing. This allowed the developers access to a small data set, used to evaluate each ISD. Evaluation results were given to the Boeing authors to ensure the data authored fulfilled requirements designed into the ABDAR Demonstration System.

NCI was responsible for ensuring sufficient data to support the field test and Demonstrations. As part of the authoring effort, NCI verified that sufficient data existed to support the damages inflicted on the aircraft. NCI also ensured the data was technically accurate and supported the ABDAR Demonstration System functionality for each ETI. In-house SMEs reviewed the data and tested its integration with the ABDAR Demonstration System. Finally, qualified Assessors and Technicians were used to perform data verification on the damaged aircraft.

Assessment System Data

Boeing developed the data needed to support the use of the WI in the constrained field test and the collection of data by CFRS needed to populate the Debrief information.

The WI package (data and application software) were developed to support the assessment of selected wire bundles in Door 6R and the Left Wing Trailing Edge area and were passed to NCI for integration into the ABDAR Demonstration System. The CFRS debrief information was collected and made available for the ABDAR Demonstration System and complied with the Interface Control Document (ICD), developed by NCI. Other system data such as the inventories of resources and A/C status were developed by NCI, as needed. The source document for the Tool and Material inventory was the TO 1-1H-39. The source documents for the other inventories were developed from the SMEs and AUG members, as a result of the data collection efforts accomplished earlier in the contract. The aircraft status information was developed as requirements were identified by the input from the AUG members, SMEs knowledge, and SW development requirements. The data was developed to be as realistic as possible without divulging any sensitive or classified information.

IPDF/PDF Technical Data

To support the field test of the ABDAR Demonstration System, 87 TOs in IPDF format were required. Boeing converted the TOs from their in-house Xyvision authoring system format to IPDF, and then processed each file separately to add links (hyperlinks). This final process was known as *linking* or *indexing*, and was the fundamental difference between PDF and IPDF files.

Boeing was responsible for the initial development of IPDF (both text and graphics) files. Boeing provided NCI a set of files on Compact Disks (CDs) equivalent in nature to the files described in the Technical Order Conversion Requirements (TOCR) standard, TM-86-01, Revision 1, 18 Oct 1996, which defines the requirements for PDF documents. NCI expanded upon some of those requirements, in an effort to make the documents easier to use. The amount of data (files) was determined by identification of the potential damage sites and the planned damages contained within each. This amount equated to the approximately 90 volumes of paper TOs that were needed to support the field test and demonstrations. Additional linking enhancements were accomplished by NCI along with the integration and testing needed prior to the field test and demonstrations. For the authoring of the data (Indexed) in PDF, Boeing and NCI used identical sets of software applications (Adobe Acrobat 3.0 and AlliantTechSystems InfoLinker). Both organizations using the same software applications facilitated transfer and resulted in no loss of efficiency.

TM-86-01 specifies that each entry in a document's Table of Contents must be linked to its target page, and that any of those references appearing in the body of the document must be linked in a similar manner. For the ABDAR Demonstration System, *all* references in the document body that pointed to locations within the document were linked.

TM-86-01 specifies, "...the frame around source links shall be invisible". When a document was linked in this manner, the user had no visual indication of the existence of a link. The only way to see that a link existed was to run the cursor over the

suspected link. If the normal cursor became a hand with a pointing finger, a link existed which was activated by clicking it. The drawback to this method became clear when using a PMA to display the IPDF data. Very often, it was difficult to get the cursor into exactly the right position to activate the link, due to the size of the display and the coarseness of the cursor control. If the computer did not respond immediately, the user either continued trying to hit the right spot, or assumed the link did not exist. For the ABDAR Demonstration System, the frames around source links were colored either red or cyan.

When a document was linked, in accordance with TM-86-01, all of the links pointed to locations within the same document. For the ABDAR program, we considered an IPDF document in this state to be *internally linked*. However, it was also possible to create *externally linked* IPDF documents by linking references to other IPDF documents. For the ABDAR Demonstration System, the 87 IPDF files were interlinked to the maximum possible extent. In the finished documents, a red source link frame indicated an internal link, and a cyan source link frame indicated an external link.

Some additional linking was done to enhance navigation through the technical data. The Illustrated Parts Breakdown (IPB) index (TO 1F-15A-4-7) contained approximately 120,000 external links. This was done so that a user could search the IPB index for a part number (taken from a damaged component on the aircraft) using the Acrobat Exchange Find tool. Once the part number was found, the user could click on the TO reference link associated with the part number to go to the TO and figure where that part was depicted. Additional links were built into TO 1F-15A-39 to greatly simplify navigation. For example, locating damage information on a specific component in TO 1F-15A-39 required the user to navigate through two to five figures or tables. When linked according to TM-86-01, there are no links between the various figures or tables.

An Adobe Acrobat plug-in called Infolinker was used to generate the hyperlinks in the IPDF files. A set of search rules was written for each IPDF file. The rules tell Infolinker what to look for and where to find it in the text of each document. Infolinker compares the text in an IPDF file to the hotspot descriptions in the rule file to generate a database of sources and targets. The sources are then reconciled with the targets and the resulting hotspots are loaded into the document.

Although a unique rule file was required for each IPDF file, the individual rules within each rule file were similar enough that a set of generic rule files could be used as a template to create the individual rule files. The content of the generic rule files was determined by the *type* of document that would be linked (Job Guide, IPB, Fault Isolation, etc.). Links between the Job Guides references, System Subsystem Numbers (SSSN's), figures, and tables were established. Links between the Fault Isolation manual references and its paragraphs, figures, tables, and Fault Codes were established. The rule files became unique when certain additional data was added, such as the number of pages in a document, where the Table of Contents was located within a document, etc.

Infolinker (as it existed during the time the IPDF files were being generated for the ABDAR Demonstration System) would not allow for batch processing of IPDF files. Each file had to be opened, processed, and closed individually. NCI wrote an Acrobat plug-in to allow batch processing of the 87 files required for the field test using the ABDAR Demonstration System. This allowed the process to be run at night or on weekends, and eliminated the need for having someone available to "feed the machine".

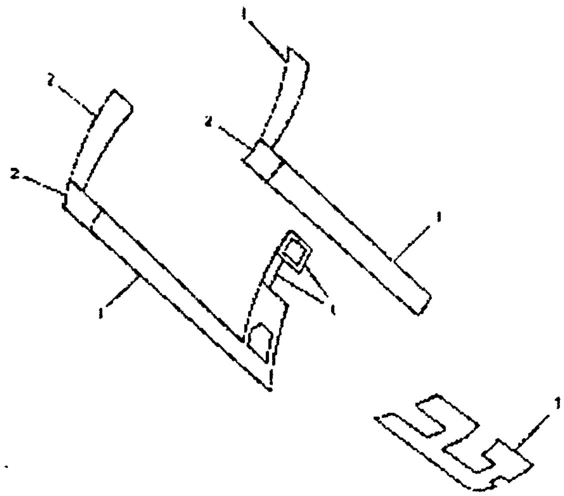
The enhancements to the IPDF files were accomplished using the approved tools (Acrobat Exchange and Infolinker). All of the links appearing in the completed IPDF files were added electronically by Infolinker without human intervention.

CDM Technical Data

Boeing provided the initial development of CDM data. NCI was responsible for the integration testing needed prior to the field test and demonstrations and any enhancements deemed necessary to support additional functionality. The CDM authoring environment that NCI and Boeing used was the QUILL Authoring System, populating a Versant database in accordance with the document type description (DTD) developed in conjunction with the Longbow Presentation System. Quill was chosen after the evaluation of available authoring systems during year one of the ABDAR program. The output of the authoring environment was SGML and CGM files capable of being parsed into the ABDAR Demonstration System Oracle database.

The current CDM MIL-STD-87269 was written to support technical data used in O-level maintenance. It was determined that MIL-STD-87269 could support technical data used during ABDR if, and only if, the data was simply being presented to the user. The attributes required by an interactive ABDR system are not currently specified in MIL-STD-87269. The majority of data required to accomplish ABDR is contained in two TOs. The General ABDR TO, 1-1H-39, contains the general information (such as documenting the AFTO Form 97), determining the class of damage(s), and making generic repairs. The aircraft specific ABDR TOs, such as the 1F-15A-39, contain specific data needed to assess each structure or system. The A/C specific TOs identify flight restrictions, coordinate locations, and possible repairs for each structure or system on the aircraft. The specific TOs use indexed graphics and tables to help the user navigate to this information (see Figure 23, Figure 24, and Figure 25 for examples of required ABDR data).

TO 1F-15A-39



IDX NO.	ITEM	MATL.	CAT	CLASS LIMITATIONS						NOTES
				A		B		C		
				K	K	K	K	K	K	
1	Skin	2024 T62	II		0.11		1.0		0.30	1 thru 4
2	Skin	2024 T81	II		0.11		[illegible]		0.30	1 thru 4

1. Repair per TO 1-1H-39.
2. Refer to chapter 1 for method of defining class limitations by means of the K factors.
3. Only one fastener out of every four may be missing. A fastener with edge distance of less than 1.5 diameters from the damage shall be considered a missing fastener.
4. All damaged area shall be covered for aerodynamics.

Figure 11-3. Structure Category Codes and Class Limitations

Figure 23 – Structure Category and Class Limitations Table

TO 1F-15A-39

Table 11-2. Avionics System Mission Serviceability Criteria

System/Subsystem/ Component	Serviceability Criteria		Remarks
	Ferry	A-A	
AUTOMATIC FLIGHT CONTROL SYSTEM (AFCS)	DP	DP	Refer to paragraph 11-11.
Accelerometer Assembly	DP	DP	Refer to paragraph 11-12 and figure 11-1.
COMMUNICATION SYSTEM	DP	DP	Refer to paragraph 11-13.
Identification Friend or Foe (IFF/UHF) Antenna	DP	DP	Refer to paragraph 11-14 and figure 11-1.

Figure 24 – System Serviceability Table

System Assessment/Repair					
Index No.	Nomenclature	Acceptable Damage	Maintenance Action/Repair Reference	Functional Check	Effects/Restrictions
	Automatic Flight Control System (AFCS)				
1	Accelerometer Assembly	See note 1.	Replace per (22-10-20) or deactivate per note 2.	Check per (22-10-20)	If deactivated, CAS system will remain functional. Up and away flight will be less sensitive in pitch and will be easy to over-g the aircraft for large stick inputs. The nose of aircraft will tend to wander in pitch and yaw during flight.
	Communication Systems			Do a BIT check.	
2	Lower Identification, Friend or Foe (IFF)/UHF Antenna	See note 1.	Replace per (23-[illegible]-20), or (23-[illegible]-20), or (23-[illegible]-20), or (23-[illegible]-20), or deactivate per note 2.	Do a UHF No. 1 radio check using lower UHF antenna.	If damaged, IFF transponder system and UHF communication receiver-transmitter No. 1 may be degraded.
1. Any damage that does not restrain the function of the component. 2. If equipment is damaged and is not going to be replaced, deactivate by removing all interfacing parts. Stow wire bundles and tape connectors. Pull circuit breakers unless other components are affected.					

Figure 25 – System Assessment/Repair Table

The interactive approach, as demonstrated in the CDM concept, required the individual data to be identifiable and the application to interface the data with the electronic collection form. There are several approaches one could take to accomplish this interaction. To prevent the need for modification of MIL-STD-87269 and existing authoring systems, the creators could develop a consistent structured CDM -- System -- Descriptive Information -- Table Element for each type. By developing a consistent structure that provided this information and having an application that it could interface with, the data collection form could be populated. NCI took this approach to prove the concept. This gave the flexibility needed to add or modify data and the application late in the ABDAR program with minimum impact.

A non-interactive approach would be to display the CDM -- System -- Descriptive Information -- Table Elements and allow the user to populate the AFTO Form 97 with the appropriate information. This CDM approach would be very similar to the IPDF concept demonstrated in this project.

Another approach would be to modify MIL-STD-87269 to support the needed data for ABDR. This would require, as a minimum, that the following specific data be made available for each CDM system element. Each CDM system element would need a type (Structure, System, or Wiring) to identify the form needed for data collection, and based on type, a set of attributes/elements to contain the following information needed to evaluate and document the damage.

a. A 'Structure' CDM system element would then need to make available its category of structure, the type of material it is made from, notes, comments, and

restriction applicable, and limits for each class of damage. These limits could be interactive where the user inserts the size and the application compares it to the limits and identifies the class and the possible repairs. Alternatively, they could be presented to the user where he does the comparison of actual size of damage to the limits established.

b. A 'System' CDM system element would carry its serviceability criteria for each mission the weapon system could fly, the restrictions if the system is not repaired, evaluation limits, work unit code and reference designator (if applicable), applicable notes and comments, and be able to identify all the tasks associated with evaluating, deactivating, and repairing the system.

c. A 'Wiring' CDM system element would carry the serviceability criteria and restrictions for the A/C systems that are contained in the bundle, the wire bundle number (for initializing the digital wiring tool), and be able to identify all tasks for repair or deactivation.

d. Also needed would be a method to view the general TO 1-1H-39 information at any given time. While the current authored documentation section of the TO 1-1H-39 is provided for the paper AFTO Form 97, this would need to be rewritten to support the electronic documentation associated with an ABDAR System. The repair tasks contained in the TO 1-1H-39 would be developed as CDM tasks and the application should allow for User modifications for these tasks that are identified as possible repairs for a damaged entity.

This section, Data Development, presented the processes necessary to develop data for execution of the ABDAR Demonstration System. The data was subdivided into three types, Assessment System, IPDF/PDF, and CDM. The next section, Field Test Overview, presents a summary of the purpose, goals, and conclusions of the field test using the ISD #4 ABDAR Demonstration System.

FIELD TEST OVERVIEW

The field test using the ABDAR Demonstration System provided a structured approach to the testing of the ABDAR Demonstration System. The field test occurred at Robins AFB between September 1998 and October 1999. The test consisted of two distinct phases. In Phase I, data was collected on performance of the ABDAR process using Paper Technical Data. In Phase II, data was collected on performance of the ABDAR process using the ABDAR Demonstration System and electronic technical data (CDM and IPDF). The specific goals of the field test were:

a. To determine if use of the ABDAR Demonstration System improves the speed, accuracy, and completeness of the ABDR process.

b. To test the ABDAR Demonstration System in a simulated ABDR environment.

c. To collect data that clearly demonstrates the advantages and disadvantages of the ABDAR Demonstration System for supporting the damage assessment process, when compared to the current paper based method.

d. To identify changes necessary in the ABDAR SSS to provide the basis for development of the most effective battle damage assessment system for operational use.

The field test compared three Media types (CDM, IPDF, and Paper) and two technician types (F-15 Assessor and Other Assessor). Dependent variables of Speed, Accuracy, and Completeness were measured. Subjects using the ABDAR Demonstration System with CDM data performed significantly faster than subjects using the ABDAR Demonstration System with IPDF data, improving the overall time by 86%. Subjects using CDM and IPDF data were significantly more accurate and complete than subjects using Paper, regardless of Technician Type. Subjects using CDM were 39% more complete and 51% more accurate than subjects using Paper. Subjects using IPDF were 34% more complete and 44% more accurate than subjects using Paper. Overall, the ABDAR Demonstration System tools, in conjunction with electronic technical data, provided a significant advantage over the current, paper-based method of performing ABDR.

In addition to the demonstrated performance enhancements to ABDR, the ABDAR Demonstration System has a high rate of acceptance among the potential users along with a strong desire to see it implemented. The field test, using the ABDAR Demonstration System, led to three recommendations by the ABDAR team.

a. That an ABDAR System be implemented for USAF ABDR personnel.

b. That efforts be made to improve the speed with which assessors use IPDF media with an ABDAR System by improving the IPDF data type and IPDF user-interface.

c. That future weapons systems use CDM data from the beginning of the program.

For detailed information regarding the field test, see Volume 3, Field Test Report.

CONCLUSIONS, LESSONS LEARNED AND RECOMMENDATIONS

The ABDAR Demonstration System demonstrated that ETI data could effectively be used as an assessment aid. Therefore, AFMC should continue its efforts to leverage the Joint Computer-Aided Logistics Support (JCALS) initiative to convert paper TO data into electronic format. Additionally, a generic assessment tool should be developed to aid the end user in collecting data at the aircraft while using the electronic data.

Requirements Analysis Phase

This phase of the ABDAR Demonstration System development program was successful in identifying and describing the requirements for the system. The IDEF models provided a detailed description of the functions and processes employed by ABDR personnel. The requirements documented in the initial SSS provided a baseline that the team used to monitor the progress of the ABDAR Demonstration System throughout development. A set of prioritized user needs were documented and satisfied and requirements traceability was maintained. An updated SSS captured all of the requirements implemented in the ABDAR Demonstration System.

a. Knowledge gained during requirements analysis:

1. Interview Methods. Two basic interview methods were used during early data collection. In the first method, the interviewer assumed no knowledge of the ABDR process. The interviewer would ask the subject to describe their job with little to no prompting other than to ask, "what do you do next?" During the second method, the interviewer would provide a framework, in the form of an ABDR process flow chart, for the subject while asking the questions. The interviewer asked general questions about the overall process and then detailed questions about each function within the flow chart. This allowed the interviewer to establish credibility by displaying knowledge of the ABDR environment.

   RECOMMENDATION: Any team performing a detailed analysis of a process should have at least a cursory understanding of the system before undertaking the data collection process with the users.

2. IDEF and UML. The IDEF0 model is a functional breakout, the IDEF 3 is a process flow, and the IDEF 1X is an entity-relationship diagram (See Figure 26). These models would have provided utility in a process-oriented design methodology. However, an object-oriented design approach was used for development of the ABDAR Demonstration System. The Object Oriented (OO) design approach was dictated by the choice of programming languages. The original SOW directed the use of Ada. This was later replaced by C++ and finally Java. All of these languages are OO languages. To translate the IDEF models into an OO design methodology, the IDEF 3 models had to be converted to use-cases and the IDEF 1X were used to generate concept models (the IDEF 0 provided little value). ABDAR development would have been more efficient if it had used an OO methodology such as Unified Modeling Language (UML). The UML process is much more conducive to OO design and development.

   RECOMMENDATION: Use of UML in OO projects.

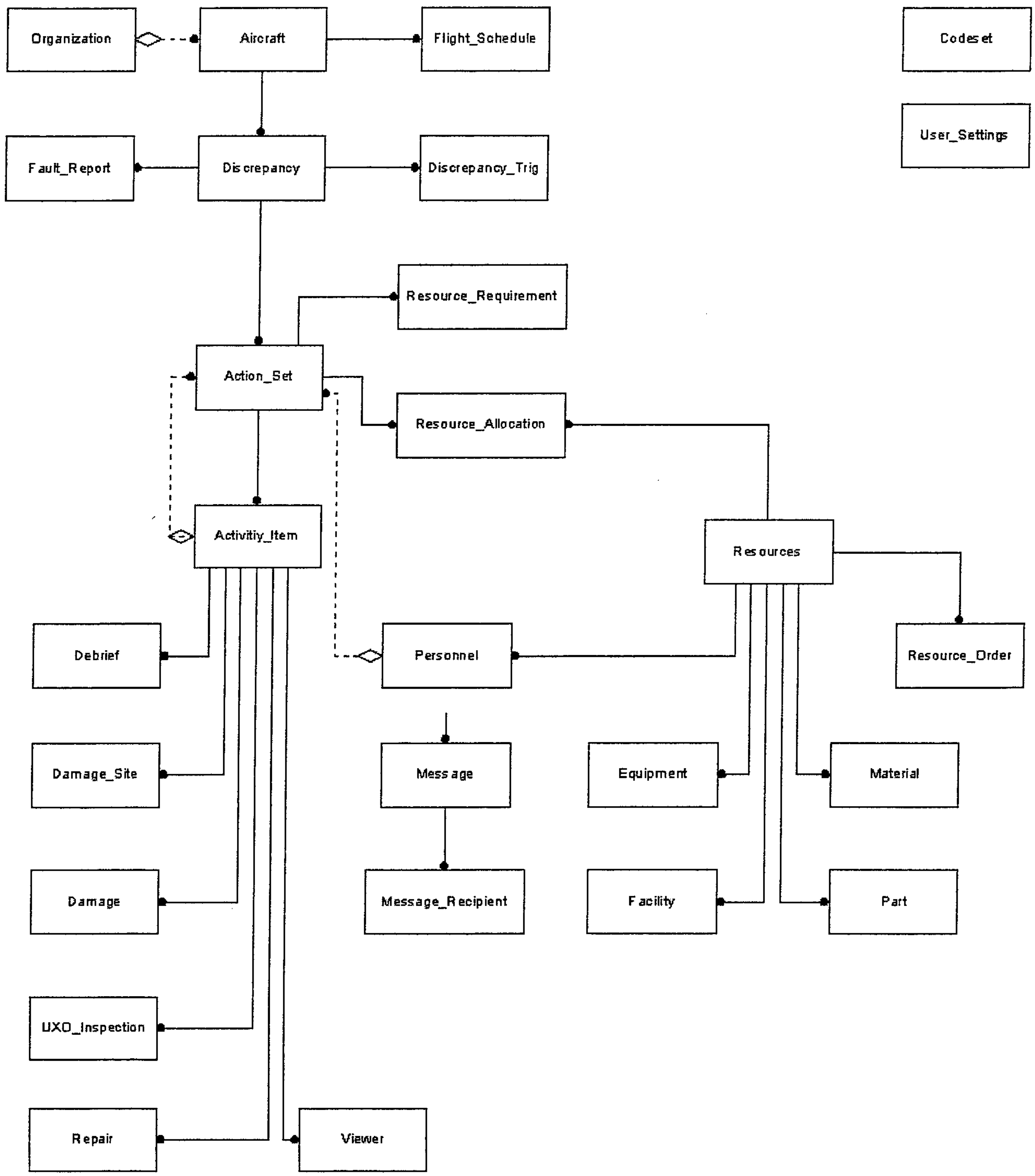


Figure 26 – ABDAR Entity-Relationship Diagram

Software Design and ABDAR Demonstration System Development Phases

These two phases provided the framework necessary to develop an ABDAR Demonstration System that met the user's needs and the field test requirements. The prototyping methodology provided the flexibility necessary to address new technologies and supported user involvement throughout the entire program. This resulted in an ABDAR Demonstration System that met user's needs, both in the stand-alone mode and in the connected mode, and contained both IPDF and CDM technical data.

a. Knowledge gained during software design and development:

1. Use of Java as a development language. Enterprise JavaBeans (EJB) is a specification and method for developing components that reside on a server. The goal of the EJB specification is to define a standard for creating components that participate in distributed OO applications. By using EJBs, developers are able to quickly build distributed programs by integrating targeted, well-defined pieces. Despite some limitations, the EJBs effectively delivered the functionality needed to develop the ABDAR System. A single EJB requires the instantiation of at least three objects (an interface, a finder, and the bean). This overhead increased the time required saving assessment information to the database. In most information systems system, slowdowns become apparent when large amounts of data are being stored or retrieved. However, with EJBs the slowdown is due to the level of object granularity specified by the EJB architecture. The ABDAR design mapped each entity in the database directly to its own EJB (the recommended design when using entity or persistent EJBs). The design is an example of fine granularity in OO development. Fine granularity is an effective way to design a system, if the objects map consistently to a unit of work. The ABDAR System did not have consistent units of work. For example, in Damage Collection the assessor used one "action" at a time, as opposed to Repair Planning, which allowed the assessor to open and manipulate all the actions simultaneously. Sun Microsystems is aware of these limitations and is currently making changes to the EJB specification to rectify this issue. Changes made to the EJB1.1 and JDBC 2.0 specification will allow for more effective set manipulations. These changes would have greatly affected the performance of the ABDAR design.

   RECOMMENDATION: When these standards are supported by application and database servers, the EJB architecture would become the architecture of choice if ABDAR is developed as an enterprise application.

2. Use of a three-tier architecture. The ABDAR Demonstration System was developed as an enterprise application. An enterprise application is generally viewed as a real-time program that supports all functions and users of a specific process. For example, the ABDAR Demonstration System supports the assessor while identifying damages, the team chief in scheduling maintenance tasks, the resource manager in ordering and inventorying

material, etc. An enterprise solution was appropriate given the requirements identified in the SSS, however, these types of applications are currently expensive to develop and maintain. The ABDAR Demonstration System's biggest positive impact was in the area of damage collection. Since the damage collection functionality does not indicate an enterprise solution, the simplest, most cost-effective way, to quickly improve ABDR would be to develop a non-enterprise application that supports damage collection. This system should be developed with hooks, so that it can be expanded to the enterprise when the costs become less prohibitive (over 70% of the ABDAR requirements were outside the damage collection area).

RECOMMENDATION: Develop a system to support damage collection, and set long-term goals to move the application to the enterprise.

3. Use of the ABDAR Users Group (AUG). The AUG allowed the users to witness the result of their input and participation in the ABDAR system, during development. Due to their active participation, the users became advocates of the system. In addition to the formal AUG process, informal communication and comments were common among the users. Overall, the enthusiasm from AUG members contributed significantly to product validity, usability, and acceptance.

   RECOMMENDATION: Users Groups should be an integral part of future software development efforts.

4. Adding to the AUG process. AUG commentary, both positive and negative, provided invaluable feedback to the ABDAR team as the system evolved. The size of the AUGs, however, should be better controlled. The UML methodology recommends approximately seven people per session. Many of the ABDAR AUG meetings were attended by more than 20 people. This environment forced a less "hands-on" approach than was desired. The AUG members would observe the presentation and take notes or initiate discussion on specific items. Although this process worked quite well, it could have been improved. If smaller groups were used, additional user testing could have been performed earlier in the development. Hands-on user testing gives the users a better feel for the utility of the system than a presentation.

   RECOMMENDATION: Create early opportunities for user testing and provide opportunities throughout the development cycle.

5. IDEF Methodology vs. Prototyping. The ABDAR Demonstration System was developed using a prototyping development methodology. Usually, when a prototyping methodology is used, requirements analysis occurs throughout each development cycle. The requirements analysis approach used on the ABDAR project was similar to one that would be used in a waterfall methodology for system development. The problem with the waterfall approach is that the requirements analysis phase quickly begins to produce

diminishing returns. As more subjects are interviewed, the information learned from each subject decreases. The ABDAR project should have begun prototype development as soon as there was a basic understanding of the ABDR process. The prototypes would then have been used for further analysis. The amount of time spent refining the requirement list and documenting the ABDR process in the IDEF models provided minimum benefit.

RECOMMENDATION: Use of prototypes instead of IDEF models in future development efforts.

6. Iterative development. The iterative process for developing the software was quite successful. The "concept, develop, demonstrate, analyze, and repeat" method worked. However, it can be improved. A more formal process that includes specific schedules and more internal code and design reviews would be more efficient.

   RECOMMENDATION: Use a more formal process that includes specific schedules and more internal code and design reviews. This approach would be more efficient.

7. Non-data type specific application. The ABDAR Demonstration System successfully proved that a technical data system could be developed using different data types (PDF and CDM). Unfortunately, the system was tightly coupled to the viewer of the respective data types and would not work if Adobe Acrobat or the in-house CDM viewers were replaced with different, but comparable, applications. To integrate a new viewer into the ABDAR Demonstration System would require an extensive programming effort. Supporting other viewers was not a requirement of the ABDAR Demonstration System because it was developed to demonstrate a specific concept. However, a production level system should support other viewers. Currently, many airframes have data stored in SGML format with various DTDs. Similar types of commercial data are often stored in HTML or XML format. Since each of these data types are associated with a specific viewer, it would be prudent to develop an ABDAR System that easily integrates whatever viewer is currently popular. To effectively reduce the effort of integrating and supporting multiple viewers, an API should be developed for a production level ABDAR System. The API would provide accessibility to the internal ABDAR System function. Assuming the viewer has an API (most COTS applications do), a DDE component can be developed to provide communication between the applications that use both APIs. Therefore, any viewer can be plugged into the ABDAR System simply by developing a single DDE component.

   RECOMMENDATION: Develop and document a robust, well defined, API for the ABDAR System that provides visibility into the internal system functionality and allows for the easy integration of data-type specific viewers.

ABDAR Demonstration System Implementation and Field Test Phase

This phase of the program focused on the field test needs and continued to evolve the ABDAR Demonstration System. The system utilized in the field test met the objective of increasing the speed, accuracy, and completeness of the ABDR assessment process. By meeting the objectives, the overall ABDR process was improved. The successful field test validated the ABDAR requirements, and provided transition agents such as AFMC/LGX and the Weapon System Program Offices with justification to continue developing advanced assessment capability. Complete discussion of the field test planning, execution, and results can be found in Volume 3 of the final technical report.

a. Knowledge gained during implementation and field testing:

1. Field Test Planning. The field test was well planned, executed, and produced significant results demonstrating the desired objectives of the research. Two factors that contributed to success were the early identification of demonstration requirements and the scheduling of a pre-field test exercise. The scheduling of the pre-field test exercise at Robins AFB, GA assured all necessary actions had been taken and coordinated within the field test team.

   RECOMMENDATION: Future AFRL programs identify testing requirements as early as possible and conduct a pre-field test to identify problem areas.

2. The ABDAR wizards. The final implementation of the system included the addition of wizards. The need for wizards was identified late in the development process during user testing. Essentially, the wizards served two purposes: training the users on the system and the ABDR process; and training the users on the use of PDF data. During the field test they proved invaluable as aids in using the system.

   RECOMMENDATION: Develop wizards for training and novice users of future software systems.

The intention of these conclusions, lessons learned, and recommendations is to share the experiences and wisdom gained during the five years of the ABDAR effort. Hopefully, the knowledge gained and shared can contribute to the success of programs that will have the benefit of learning from the ABDAR effort.

REFERENCES

Department of Defense. (1992). *Military Specification: Database, Revisable: Interactive Electronic Technical Manuals, for the Support of* (MIL-D-87269). Philadelphia, PA: Naval Publishing and Printing Service Office.

Department of Defense. (1992). *Military Specification: Manuals, Interactive Electronic Technical: General Content, Style, Format, and User-Interaction Requirements* (MIL-M-87268). Philadelphia, PA: Naval Publishing and Printing Service Office.

GDE Systems. (1995). *Integrated Maintenance Information System: System Segment Specification* (Unpublished manuscript). Wright-Patterson AFB, OH: Logistics Research Division, Human Resources Directorate, Armstrong Laboratory.

General Dynamics Electronics Division. (1990). *Integrated Maintenance Information System (IMIS) Phase II Architecture*. (Unpublished manuscript). Wright-Patterson AFB, OH: Logistics Research Division, Human Resources Directorate, Armstrong Laboratory.

NCI Information Systems, Inc. *Aircraft Battle Damage Assessment and Repair: Assessment Logic* (Unpublished manuscript). Wright-Paterson AFB, OH: Deployment and Sustainment Division, Human Effectiveness Directorate, Air Force Research Laboratory.

Thomas, D.L. (1995) *Integrated Maintenance Information System: User Field Demonstration and Test: Executive Summary.* (AL/HR-TR-1995-0033, AD-A293250). Wright-Patterson AFB, OH: Logistics Research Division, Human Resources Directorate, Armstrong Laboratory.

Ward, G.D., Kruzick, T.M. & Weimer, R.E. (1995). *Integrated Maintenance Information System (IMIS) Final Program Report, Volume 3: Results, Conclusions, and Recommendations* (AL/HR-TR-1995-0035, AD-A302291). Wright-Patterson AFB, OH: Logistics Research Division, Human Resources Directorate, Armstrong Laboratory.

Ward, G.D., Kruzick, T.M. & Weimer, R.E. (1995). *Integrated Maintenance Information System (IMIS) Final Program Report, Volume 1: Executive Summary* (AL/HR-TR-1995-0040, AD-A303609). Wright-Patterson AFB, OH: Logistics Research Division, Human Resources Directorate, Armstrong Laboratory.

Ward, G.D., Kruzick, T.M. & Weimer, R.E. (1995). *Integrated Maintenance Information System (IMIS) Final Program Report, Volume 2: Program Methodology* (AL/HR-TR-1995-0041, AD-A303608). Wright-Patterson AFB, OH: Logistics Research Division, Human Resources Directorate, Armstrong Laboratory.

Wilper, B.L., Eschenbrenner, A.J., Jr. & Payne, W.T. (1983). *Assessing Aircraft Battle Damage: Information and Support Requirements* (AFHRL-TR-83-25, AD-B078865). Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

ACRONYMS

ABDAR	Aircraft Battle Damage Assessment and Repair
ABDR	Aircraft Battle Damage Repair
ACC	Air Combat Command
AF	Air Force
AFB	Air Force Base
AFIT	Air Force Institute of Technology
AFMC	Air Force Materiel Command
AFMC/LGX	Air Force Materiel Command/Logistics Plans and Programs
AFRL	Air Force Research Laboratory
AFRL/HESR	Air Force Research Laboratory/Deployment and Sustainment Division, Logistics Readiness Branch
AFSOC	Air Force Special Operations Command
AFTO	Air Force Technical Order
ALC	Air Logistics Center
AMC	Air Mobility Command
API	Application Program Interface
AUG	ABDAR Users Group
BA	Bundle Assessment
CAF	Combat Air Forces
CAMS	Core Automated Maintenance System
CDM	Content Data Model
CD	Compact Disk
CFRS	Computerized Fault Reporting System
CLSS	Combat Logistics Support Squadron
CORBA	Common Object Request Broker Architecture
DDE	Dynamic Data Exchange
DoD	Department of Defense
DTD	Document Type Description
DTIC	Defense Technical Information Center
DWDS	Digital Wiring Data System
EJB	Enterprise Java Bean
EOD	Explosive Ordnance Disposal
ETI	Electronic Technical Information
ETTR	Estimated Time To Repair
GUI	Graphical User Interface
HQ	Headquarters
HTML	HyperText Markup Language
IADS	Interactive Authoring and Display System
ICD	Interface Control Document
ID	Identifier
IDEF	Integrated Computer Aided Manufacturing (ICAM) Definition Methodology
IETM	Interactive Electronic Technical Manual
IMDS	Integrated Maintenance Data System
IMIS	Integrated Maintenance Information System
IMISA	IMIS Architecture
IPB	Illustrated Parts Breakdown
IPDF	Indexed Portable Document Format

ISD	Interim Software Demonstration
IT	Information Technology
JCALS	Joint Computer-Aided Logistics Support
JDK	Java Developer's Kit
LAN	Local Area Network
MDA	McDonnell Douglas Aerospace
MESL	Mission Essential Subsystem List
MIL STD	Military Standard
NT	New Technology
OO	Object Oriented
PC	Personal Computer
PCMCIA	Personal Computer Memory Card International Association
PDF	Portable Document Format
PFN	Process Flow Network
PL/SQL	Procedural Language extension to Structured Query Language
PM	Program Manager
PMA	Portable Maintenance Aid
PMO	Program Management Office
PMR	Program Management Review
R&D	Research and Development
RF	Radio Frequency
SDK	Software Development Kit
SGML	Standard Generalized Markup Language
SME	Subject Matter Experts
SOO	Statement of Objectives
SOW	Statement of Work
SPO	System Program Office
SSS	System/Subsystem Specification
SSSN	System Subsystem Number
SURVIAC	Survivability and Vulnerability Information Analysis Center
TO	Technical Order
TOCR	Technical Order Conversion Requirements
TTP	Technology Transition Plan
UML	Unified Modeling Language
USAF	United States Air Force
UXO	Unexploded Ordinance
VGA	Video Graphics Array
WBS	Work Breakdown Structure
WI	Wiring Illuminator
WPAFB	Wright-Patterson AFB
WWW	World Wide Web

Appendix - Aircraft Battle Damage Assessment and Repair (ABDAR) Interim Software Demonstration ISD #5 Concept Analysis Paper

1. Introduction

With Government discussions and participation, Interim Software Demonstration (ISD) #5 was approved as a significant departure from previous ISD's. The Government determined that ISD #5 not be prototyped and be non-iterative. The Government wanted to develop ideas and thoughts in a creative manner, with a goal of improving Aircraft Battle Damage Repair (ABDR). ISD #5 did not necessarily have to use the current ABDAR Demonstration System. The **ABDAR** team, with Government participation, held a series of design team meetings to discuss and review ideas.

2. History of ISD #5

a. ISD #5 is unique from the previous ISD's and ISD #6. ISD #5 is not an iterative step in the development of the ABDAR Demonstration System; rather, it is a collection of thought-provoking ideas to enhance the ABDR assessment process. As such, the processes behind its genesis differed significantly from previous ISD's. The ABDAR Team, with Government participation, invested a significant amount of creative effort in developing ISD #5. This paper documents the ideas and results produced during the effort.

b. In an attempt to redefine the Aircraft Battle Damage Repair (ABDR) process, the **ABDAR** team decided to view the process as a genesis of ideas. The team worked cooperatively and quickly to develop as many ideas as possible for re-defining performance enhancements for ABDR. The team documented and discussed each idea for a brief amount of time. After discussion, the team decided whether the idea warranted further detailed discussion and acted accordingly.

c. The team developed 13 ideas. Some ideas concentrated on the use of tools or technology to help ABDR, while others examined the benefits of enhancing the current **ABDAR** Demonstration System. A combination of several of the 13 original ideas culminated in the Remote Assessment Pilot Study being selected as the initiative with the most merit.

d. The following section details all of the candidates developed by the team. Each idea includes a summary, team assessment, and possible next steps.

3. ISD #5 Candidates

3.1 ABDAR Trainer

a. Summary: Make the ABDAR Demonstration System a training device for technicians and assessors.

b. Team Assessment: Current ABDR training is hands-on using the paper technical orders (TO's). Hands-on training during an exercise is expensive and inefficient,

particularly for inexperienced assessors. Manual TO training is abstract and transfers poorly to an actual assessment. **ABDAR** allows for detailed corrections and teaching during the scenario (as opposed to exercises) and provides a sort of realism that cannot be achieved by reading the TO.

c. Next Step: Use the ABDAR Demonstration System to bridge the gap; consider it a virtual trainer that simulates the ABDR environment.

3.2 ISD #1 Redux

a. Summary: Present a focused, scaled-down demonstration that highlights the portions of the system that proved to be most beneficial during the field test.

b. Team Assessment: Highlighting the most efficient portions of ISD #4 will permit giving concrete examples of improvements.

c. Next Step: Having the back end stripped out will be a true demonstration of the ABDAR Demonstration System. It should also implement "high-tech" solutions to specific problems.

3.3 Borescope System

a. Summary: A borescope could be used to view hidden damage(s) without having to remove components or cut away structure for access purposes.

b. Team Assessment: Would be advantageous to use for Unexploded Ordnance (UXO) inspections by looking in the entrance hole instead of having to pull panels (dangerous if UXO is present.) Some examples are that a technician can take measurements, allowing the "Class" of damage identification to be made quicker. In addition, the borescope system has uses in exploring composites and Low Observable (LO) structure to measure the hidden damages that radiate outward from entry impact.

c. Next Step: Acquire a borescope system to interface with the ABDAR Demonstration System to demonstrate the benefits it would provide.

3.4 Robotic Assessment

a. Summary: Program a robot to travel out to the aircraft, take pictures of the aircraft, and send these pictures back to the assessor real-time. If holes are spotted, the assessor could direct the robot, via radio control, to inspect holes further by taking more pictures and perhaps using a small, fiber optic camera to go inside the hole. After identifying the extent of the damage, the assessor can then evaluate a repair sequence path to affect timely repairs.

b. Team Assessment: By measuring the damage size, noting the position of the damage, and comparing it to the information in the TO or electronic data, the assessor can determine the "Class" of damage. If this could be done electronically with "before

and after comparisons", speed and accuracy should increase. Additionally, checking for UXO with a robotic system has obvious safety benefits.

c. Next Step: The purchase and programming of an off-the-shelf robotic kit to perform the necessary functions. Development costs may be excessive.

3.5 Patch Generator

a. Summary: There are common types of repairs for fixing structural damages found in ABDR (i.e., Double Row Patch, Single Row Patch, Step Patch, Extruded Angle Single Flange, Honeycomb, etc.). Development of a software tool, that allows assessors to select from a list of these common repairs and modify them as needed, is beneficial. Some of the obvious operations would be: change the size of the patch, change the row pitch, change the material type and thickness, add rivets to pick up existing holes, and add external stringers.

b. Team Assessment: All these modifications could be performed visually using mouse point, click, and drag operations.

c. Next Step: Implementing and integrating this idea with the existing ABDAR Demonstration System.

3.6 Digital Camera

a. Summary: Currently, the assessor describes damages verbally for individuals not able to view the damage. The assessor needs to be explicit enough for those not present to get an understanding of the extent of the damage. Explicit detail is also required for the documentation of the repair accomplishment. When communicating with the Engineer or Original Equipment Manufacturer (OEM), pictures of the damage have obvious benefits.

b. Team Assessment: Attaching digital pictures to damages, repairs, communications, and documentation is an efficient and effective method for assessors and engineers to communicate.

c. Next Step: Interface a digital camera system with the ABDAR Demonstration System and demonstrate the advantages it provides to all aspects of ABDR.

3.7 GroupWare

a. Summary: The ABDAR Demonstration System stores data in a database, but provides little output for higher-level decision making. GroupWare would turn the ABDAR process into more of a collaborative environment by using technologies such as: e-mail, group calendars & scheduling, real-time or near-real-time laptop conferencing, group document handling, shared white board, and a live help desk. Other ideas include shared Internet white board and video conferencing tools, and replacing the ABDAR tools entirely with HyperText Markup Language (HTML) forms that can be accessed via the Internet.

b. Team Assessment: GroupWare could be implemented with a limited set of current ABDAR tools and third party off-the-shelf software like Outlook (which provides email, group calendars, and resource scheduling).

c. Next Step: Further evaluation recognized that the GroupWare idea moves ABDAR back to a web browser application.

3.8 Remote Assessment

a. Summary: A non-assessor, using a digital camera or digital video camera, collects data about an aircraft's damage(s) and forwards [the data] to trained Assessors or Engineers, not located at the aircraft site, to evaluate the collected data. This digital information, along with size and location, is sent to the remote assessor.

b. Team Assessment: The assessor uses available resources to accomplish the assessment and identify the repairs. It needs to be determined if digital information is adequate to make effective repair plans.

c. Next Step: Researching this idea along with the digital camera (candidate 3.6).

3.9 Mini-Apps

a. Summary: Identify the problem spots in ABDR where the technician benefits most from automation. Some likely areas are: mathematical operations, sorting tasks, identifying scheduling conflicts, etc. Then provide a specific tool for each of the identified areas.

b. Team Assessment: Unlike the Wizard idea, no "step-by-step" help is given. Technicians, properly trained to use the tools, will automatically go to the tool for assistance.

c. Next Step: Identify areas in the ABDR process where automation would be beneficial.

3.10 Extended Markup Language (XML) Conversion

a. Summary: One of the technology areas worth exploring in ISD #5 is the implementation of the Content Data Model (CDM) with XML. CDM currently uses SGML; XML is a subset of SGML.

b. Team Assessment: XML is simpler than SGML and may make the authoring and presentation of CDM data an easier process.

c. Next Step: XML needs more research to prove to programs (such as the F-22) that they are not obsolete with CDM data. ABDAR presents an opportunity to perform some of the preliminary work. Performing a mapping of CDM elements to XML, as well as developing a simple viewer in Java to be integrated into the ABDAR Demonstration System, or demonstrated in a stand-alone fashion.

3.11 Palm Pilot Assessment Tool

a. Summary: Provide the assessor with a very simple tool, such as a Palm Pilot appliance. The palm appliance would help the assessors in collecting data about damage(s) while at the aircraft. The tool could help by providing simple palm applications that work with a fully implemented ABDAR Demonstration System.

b. Team Assessment: Palm applications are small and easy to use. The detailed inspection tool would prompt the user while at the aircraft to collect just the basic information required for each major damage type. This information could be integrated with the full ABDAR Demonstration System to do research the TO database.

c. Next Step: Integrate this concept with the Digital Camera and explore the Remote Assessment concept as a possible change to the current ABDR process.

3.12 3D/Virtual Reality

a. Summary: Use of a 3-D/Virtual Reality (VR) model would allow the assessor to "be" the projectile and preview the path it most likely would have taken.

b. Team Assessment: With this preview, the assessor could better determine the actual path. Potential training uses also exist.

c. Next Step: Explore Boeing's effort in this field. Document it separately if it has merit as an ABDR capability. Consider this initiative to be a stand-alone demonstration of the concept.

3.13 Wizards

a. Summary: The current ABDAR Demonstration System demonstrates the ability to separate technical data from an ABDAR specific application. It also demonstrates that a wizard enhances the presentation and use of technical data.

b. Team Assessment: Unfortunately, the assessment logic and the system are integrated and it is difficult, or impossible, to separate them. For example, the order of the ABDAR functions is implied by the order of the pages in the toolbar or drop down menu.

c. Next Step: Creation of a set of HTML or XML pages that would walk an assessor through the entire ABDR process.

4. Concept Selected for Further Study.

a. The concept chosen for further exploration was Remote Assessment (candidate 3.8). A Pilot Study was designed to explore the feasibility of an Assessor performing an assessment from a remote position. The Assessor relies on an individual at the aircraft to perform requested functions. The requested functions would typically include taking digital pictures and recording measurements specified by the Assessor.

b. A simplified process flow of how Remote Assessment would be used is as follows:

1. Aircraft Assessor (AA) takes generic pictures and emails them to the Qualified Assessor (QA).

2. The QA makes notes on what to measure, requests other pictures, asks specific questions, etc., and sends annotated pictures back to the AA.

3. The AA "fills-in-the-blanks" on the annotated pictures and sends the package back to the QA.

4. The QA then uses the collected data to perform assessment and design a repair to be sent back to the AA.

c. The primary goal of a remote assessment is to remove the QA from danger. QA's are highly trained and very valuable assets. If the QA can be removed from a dangerous environment and still accomplish his assignments; the primary goal has been achieved. A secondary goal is to have the QA's special knowledge and skills maximized. Under certain circumstances, a QA could assess the damage on several aircraft concurrently, thus, increasing the effectiveness and benefits of the QA directly. Overall, this provides more efficient use of critical limited resources in a less austere environment.

d. Designing a simple demonstration system allows the performance of a limited study to test the remote assessment concept. Below is a description of the development system and proposed procedures for testing it.

5. The Remote Assessment Pilot Study

a. The Remote Assessment Pilot Study is deliberately meant to be a significant departure from previous ISD's concerning implementation methodology. With the deliberate departure, results from the Field Test will not be directly comparable to the Remote Assessment Pilot Study. However, it is desirable to have some idea of the efficacy of the remote assessment concept. Accordingly, the pilot study will provide an idea of how good remote assessment is at assisting in the performance of Aircraft Battle Damage Repair (ABDR). Resource constraints precluded a full test of the Remote Assessment concept. However, a limited study permits collection of data similar to Field Test data. This allows us to draw some relevant comparisons between Remote Assessment and the Field Test, without performing specific statistical tests. This means that only qualified statements about the success or failure of the Remote Assessment concept can be made, such as: Remote Assessment *seems* to perform as well as the ABDAR Demonstration System (ISD #4) on the assessment task. Further, it captures ideas for further improvement of the **ABDAR** Demonstration System for future evaluation.

b. Unlike the Field Test, which had multiple assessors and data types, this study will only use one qualified F-15 Assessor and one non-assessor-trained technician.

c. The Remote Assessment Pilot Study does not need to be complex. Use of Off-the-Shelf software and hardware will permit a full demonstration of this concept. In fact, the concept of Remote Assessment can be proven without the expenditure of any funds for software or hardware purchases. All that is needed for the demonstration and testing are two PC's, a digital camera, email software for accessing and sending email, software to view and modify digital images, and an assessor's kit. The **ABDAR** team currently possesses all of these items.

d. A basic demonstration of the Remote Assessment Pilot Study concept would include the following:

1. Producing a damage on an aircraft.
2. The AA takes pictures of the damage and emails the pictures to the QA.
   (a) Includes conducting UXO inspection and Safe for Maintenance.
   (b) Could involve using a Palm Pilot or other technology.
3. The QA makes notes on the pictures and emails back to the AA.
   (a) QA would indicate what damages to measure.
   (b) Request skin thickness.
   (c) Request other needed information.
4. The AA returns to the aircraft and collects the requested information.
5. The AA then emails the damage information back to the QA.
6. The QA uses the collected data to perform an assessment and develop a Repair Plan.
   (a) QA has access to ETM's, ideally using the relevant portions of the ABDAR Demonstration System.
   (b) QA can also request more information from AA at this point.
7. The QA sends the Repair Plan back to the AA to accomplish the repairs.

e. Performing Step 6 before the actual demonstration allows a quick presentation of the software and its capabilities. During the demonstration, the QA's comments are presented immediately, as opposed to having a QA develop them in real-time during the demonstration. In addition, the Repair Plan would already be accomplished and simply presented during the demonstration before being emailed back to the AA. The basic idea is to accomplish anything that takes a long time before the demonstration begins.

6. The Pilot Study

a. A small pilot study was conducted with an NCI staff scientist serving as the AA, and an experienced aircraft battle damage assessor serving as the QA. The QA digitally photographed a damage and used PowerPoint to indicate what information he needed. He then emailed the PowerPoint presentation to the AA who went to the aircraft and collected the needed data, entered it in the PowerPoint presentation and emailed it back to the QA.

b. Although limited in scope, this pilot study proved highly informative. The QA and AA experienced little or no difficulty communicating electronically with each other. Using unfamiliar software was the most significant problem experienced. The pilot study proves the feasibility of the Remote Assessment concept.

c. A more complete study would use five or six assessors filling the same roll as the QA. The AA position can be filled by anyone on the ABDAR Team as it takes no **ABDR** specific knowledge.

7. Conclusions

a. Performing this study in a limited manner, rather than making it a full experiment, has many advantages. It creates an easy logistics situation, as we will need only one F-15 Assessor at a time, which will make it easier for the CLSS's to support. Also, the procedures used can be more informal during testing, as rigorous control is not needed because statistical analysis will not be performed. A limited study also permits the introduction of technology changes at any time. Constraints on development time for ISD #5 may limit how advanced the demonstration is when the first subject is run. Therefore, realistic simulations will replace non-developed segments of the system. Replacing a simulated task with the actual task at any point in the study allows development to continue while conducting the study.

b. The biggest question at this point is implementation and determining what technologies to employ. As noted above, there is no need for special technology to perform either the demonstration or the study. However, a lack of technology does not make for an impressive demonstration. The critical element for the demonstration is the tool the AA uses as the damage collection device.

c. The ideas generated for improvement of the ABDR process illustrate the success of ISD #5. The merging of more than one idea into the "remote assessment pilot study' not only shows team effort and success but demonstrates a more efficient method for assessor and engineer resources.

d. The Government Program Manager and Deputy Program Manager participated in the creation of ideas and the selection and approval of the pilot study. Discussion and results of ISD #5 ideas and the Remote Assessment Pilot Study are planned to be presented at the **ABDAR** Users Group (AUG)/Demonstrations in December 1999.